AFRL-IF-RS-TR-2001-201
Final Technical Report
October 2001

NEXT GENERATION MEMS DESIGN TOOLS

Tanner Research, Inc.

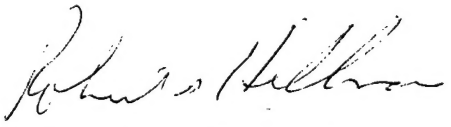
Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. E117/6220

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2001-201 has been reviewed and is approved for publication.



APPROVED: ROBERT G. HILLMAN
Project Engineer

FOR THE DIRECTOR: NORTHRUP FOWLER, III, Technical Advisor
Information Technology Division
Information Directorate

If your address has changed or if you wish to be removed from the Air Force Research Laboratory Rome Research Site mailing list, or if the addressee is no longer employed by your organization, please notify AFRL/IFTC, 26 Electronic Pky, Rome, NY 13441-4514. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

NEXT GENERATION MEMS DESIGN TOOLS

Barry Dyne

Contractor: Tanner Research, Inc.
Contract Number: F30602-96-2-0311
Effective Date of Contract: 1 October 1996
Contract Expiration Date: 31 December 1999
Short Title of Work: Next Generation MEMS Design Tools
Period of Work Covered: Oct 96 – Dec 99

Principal Investigator: Barry Dyne
Phone: (818) 792-0300
AFRL Project Engineer: Robert G. Hillman
Phone: (315) 330-4961

This research was supported by the Defense Advanced Research Projects Agency of the Department of Defense and was monitored by Robert G. Hillman, AFRL/IFTC, 26 Electronic Pky, Rome, NY.

REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	Oct 01	Final Oct 96 - Dec 99

4. TITLE AND SUBTITLE

NEXT GENERATION MEMS DESIGN TOOLS

5. FUNDING NUMBERS

C - F30602-96-2-0311
PE - 63739E
PR - E117
TA - 00
WU - 09

6. AUTHOR(S)

Barry Dyne

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Tanner Research, Inc.
2650 E. Foothill Blvd.
Pasadena, CA 91107

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFRL/IFTC
26 Electronic Pky
Rome, NY 13441-4514

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

AFRL-IF-RS-TR-2001-201

11. SUPPLEMENTARY NOTES

AFRL Project Engineer: Robert G. Hillman, IFTC, 315-330-4961

12a. DISTRIBUTION AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The objective for this project has been to lower the barriers to MEMS technology through the development of an integrated suite of CAD tools for mixed technology design. Tanner has achieved new levels of integration of layout, solid modeling, and finite element analysis for MEMS device design and verification. Tanner has developed a modeling tool that generates a solid model directly from the layout and a process description. Tanner has also developed a novel approach to MEMS device meshing, dramatically improving the efficiency of finite element analysis. System design tools developed include mixed technology schematic, simulation, place and route, extraction and design rule checking. These tools enable the design, physical layout, and validation of chips of mixed MEMS/IC components. Simulation tools allow the use of a variety of models including C-code, Spice, and MATLAB, necessary for the design of mixed technology chips. A library of components contains a schematic, simulation, and layout representation of each device, and device layout may be synthesized from functional or geometric parameters. The tools have been demonstrated at a number of commercial and government sites, and are now in commercial release.

14. SUBJECT TERMS

MEMS, micro-electro-mechanical systems, mixed-technology, EDA, modeling, finite element

15. NUMBER OF PAGES

68

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Table of Contents

List of Figures

1. Research Status

1.1. Introduction

This is the Final Report for the contract, covering new work since July 1, 1999, and summarizing the complete program since it's start date on October 1, 1996. The original end date of the contract was September 30, 1999, however an extension was granted to December 31, 1999. Contract billing reached the allocated total amount with the October 1999 bill, so this report covers all work done under the contract.

The program consists of five major task areas, i) physical design, including solid modeling, visualization, and library management, ii) finite element analysis, iii) macromodel generation, iv) schematic, simulation, and modeling, and v) layout synthesis and verification. Detailed progress on each of these areas is described below.

1.2. Physical Design

This task consists of development of a process specification, 3-D solid model generation, and rendering capability, giving the MEMS designer the ability to create and visualize the 3-D structure of a MEMS device during the design phase, prior to fabrication. The solid model is generated automatically from 2-D layer based layout data in the Tanner layout editor L-Edit, combined with a process specification. This tools provides a visual design verification aid prior to fabrication, and also provides 3D models for input to analysis tools for further verification and simulation. A designer can use this tool to ensure that the 2D layout they create has the intended 3D structure and function when fabricated. Also included in this task is component library generation and management.

1.2.1. Solid Modeling

Highlights:

- Commercial release of Solid Model module in MEMS-Pro V2.
- Developed 3D modeling and viewing tool, tightly integrated with layout editor, featuring the following capabilities:
- 3D model generation for process steps deposit (conformal, snowfall, fill), etch (bulk, surface, sacrificial), and mechanical polish.
- Checking of 3D model for synchronization with layout and process definition.
- Support of boxes, polygons, circles, arcs, tori, curved polygons, including hierarchical layout.
- Edit of process definition via a user friendly dialog, or file input
- Cross-Section viewing
- 3D model viewing with pan, zoom, rotate, spin, and color controls
- Hide layers, enabling the user to see layers that are obstructed or covered by subsequent layers.
- Display Intermediate Process Steps

- Enable/Disable Process Steps
- Export 3D model to SAT file or ANF file for transfer to ANSYS for finite element analysis
- 3D Tools menu available from context sensitive menu of the design navigator
- **Recent Highlights:**
- Implemented stretching of vertical axis in cross section rendering.
- Implemented option to export visible portion of model only, in addition to export of entire model.
- Implemented conformal deposition onto angled sidewall surfaces.

In the previous period Tanner culminated its push toward commercialization of the 3D solid modeling tools with the release of MEMS-Pro V2. Pre-release beta versions were sent to commercial sites for evaluation, including a copy sent to the contract monitor at AFRL. The final product included documentation, tutorials, examples, and technology configuration files. Details on specific features are provided below.

Database Synchronization/Integrated Layout and 3D Model

In the Tanner system, the 3D modeling tool is tightly integrated with the layout editor, allowing us to leverage a number of capabilities that would not be possible in a stand-alone tool. Figure 1 shows a view of the integrated system, with multiple windows showing the layout and 3D model simultaneously. The user can maintain a view of the entire model in one window, while zooming in on key features in another window.

A key feature of the Tanner system is that 3D models are stored directly in the layout database, allowing a 3D model to be stored for each cell in the design. This alleviates the burden from the user of keeping track of which 3D model files correspond to which layout cells. Tanner has also implemented the ability to perform synchronization checking between the layout, the process definition, and the 3D model. Upon request to view an existing 3D Model, the user will be notified if the layout or process has been modified since the 3D model was created, and given the option to view the old model or to regenerate a new one. This feature protects the user from accidentally viewing and exporting to analysis models that do not correspond to the structure that will be fabricated, and potentially from fabricating a device that fails to meet spec.

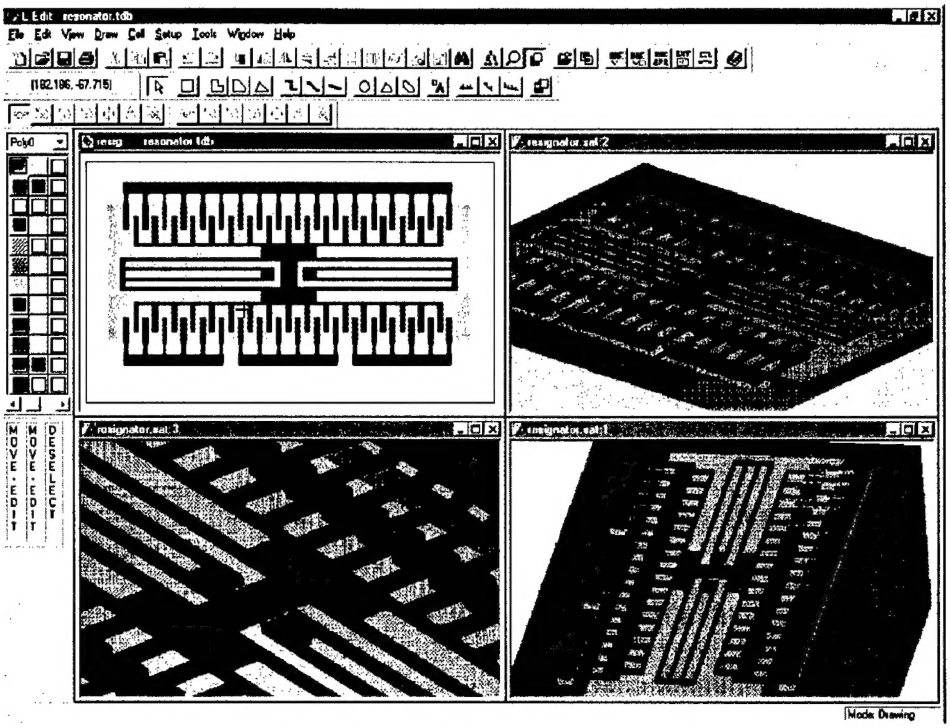


Figure 1: Solid model integrated with L-Edit.

Hierarchical Layout Support

By tightly integrating the 3D modeling tool with the layout editor, Tanner is able to take advantage of many of the features present in the layout editor. In particular, Tanner has implemented the capability in the 3D modeling engine to support hierarchical layout. Layout may contain instances of cells, rotated instances, as well as arrays of instances, and cells may be instanced, rotated, and arrayed to any level of nesting. Figure 2 shows an example of an instance of a switch, a 2x2 array of the switch, a horizontal and rotated instance, and the 3D model of the horizontal and rotated devices.

The user then has two ways to partition the design for analysis, first, he can use hierarchy to partition the design by placing components to undergo 3D modeling and analysis into separate cells. These cells then get modeled separately from the top level cell. The second approach allows the user to simply draw a box around the area to be modeled. Using this capability the user can perform 3D modeling of the device completely within the environment of the entire system design, and if necessary, surrounding circuit components may also be included in the model. The user simply draws a box on the layout identifying the region to model, and the region enclosed by the box is processed for 3D modeling. Any level of hierarchy may be present inside the box.

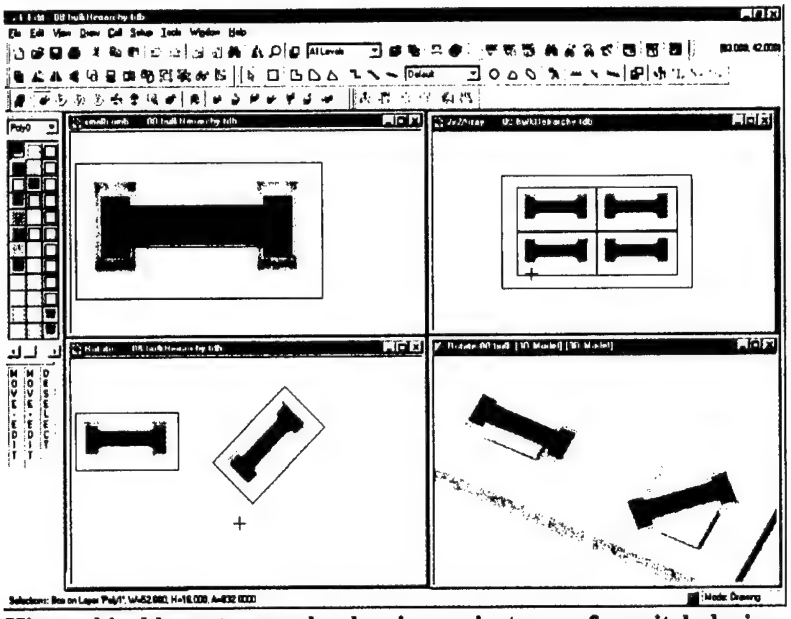


Figure 2: Hierarchical layout example, showing an instance of a switch device, a 2x2 array of the device, and horizontal and a rotated instance of the device, and the 3D model of the horizontal and rotated device.

Disable Process Steps/Display Intermediate Process Steps

In many cases, the complete fabrication process is more detailed than what is needed for 3D modeling or analysis purposes, so Tanner has implemented the ability to disable steps in the process definition. Using this feature, the user can disable certain steps to simplify the resulting model, such as removing etching of holes, or can improve performance for a given design if it is known that certain steps are not needed.

Tanner has also implemented the ability to display intermediate process steps. This enables the user to verify that the sequence of processing steps is acting correctly on the layout, and also lets the user verify that a structure does not collapse midway through processing. The feature is illustrated in Figure 3 which shows the intermediate process steps of a thermal actuator with the MUMPS process. Four intermediate steps are shown i) step8, etch anchor, ii) step 10, etch poly1, iii) etch poly2, and iv) final 3D model. After step 8 one can see the holes in the oxide layer for the anchor and dimples.

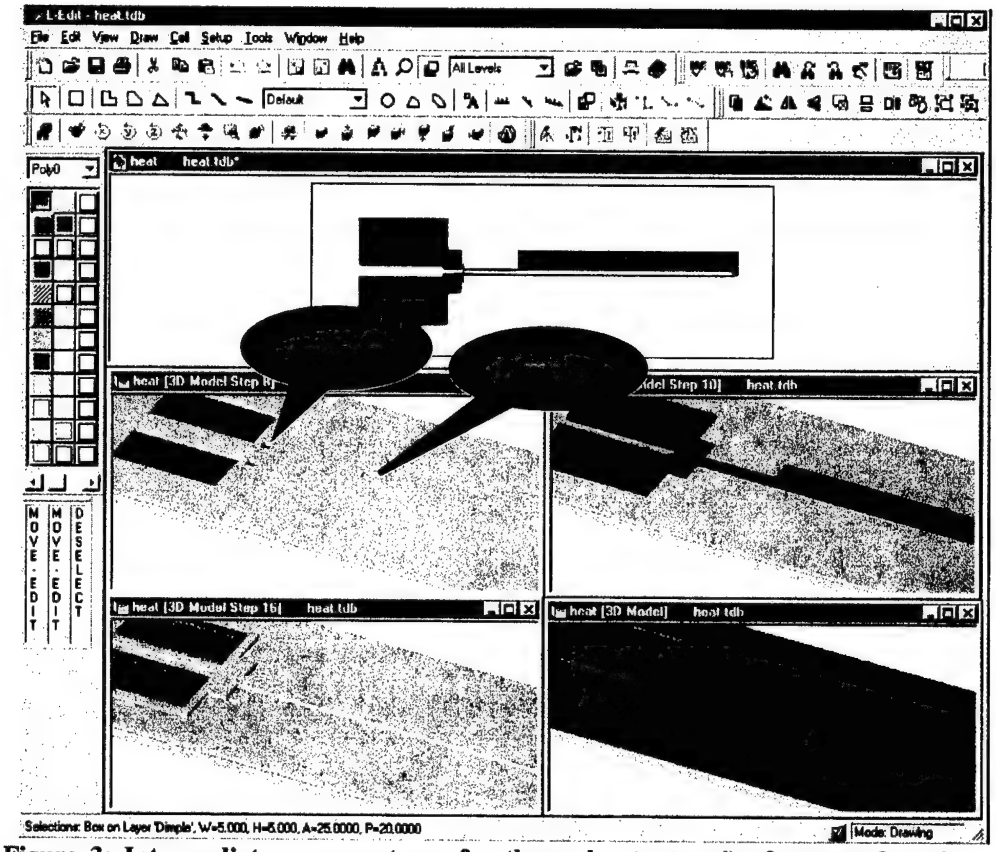


Figure 3: Intermediate process steps of a thermal actuator, i) after step 8, etch anchor 1, ii) after step 10, etch poly1, iii) after step 16, etch poly 2, and final 3D model.

Curved object support (arcs, tori, polygons)

Tanner has recently implemented curved layout drawing primitives, including arc, tori, and polygons with curved sides. Tanner has also implemented support for these structures in the 3D modeling tool. These objects are very useful in creating torsional resonator components (shown in Figure 4), motors and other devices with curved layout. Tori for example may be edited by the inner and outer radius, and start and end angle, options typical in a mechanical drawing tool but not usually found in a IC/MEMS layout tool. These parameters make manipulation of curved elements far easier than using straight line approximations as previously required. In the example below, Tanner has used another feature, the ability to rotate instances by any angle, to create different interlocking between the fixed and moving elements of the resonator.

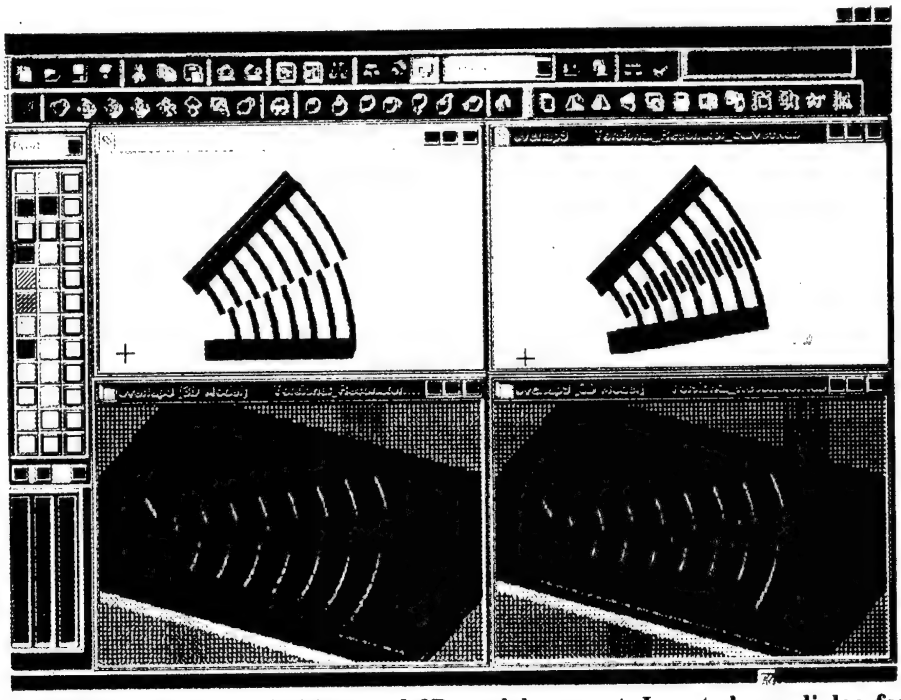


Figure 4: Curved layout primitives and 3D model support. Insert shows dialog for editing tori using Begin and End Angles, and Inner and Outer Radius.

Height Scaling

The layers from which MEMS devices are fabricated are often very thin in relation to the horizontal extent of the device. In viewing cross sections it is sometimes difficult to distinguish features. Tanner implemented the capability to scale the cross section view to either stretch, or shrink if desired, the height of the device. This is illustrated in Figure 5, which shows a cross section of a comb drive device using no scaling, and scaling by a factor of 3.

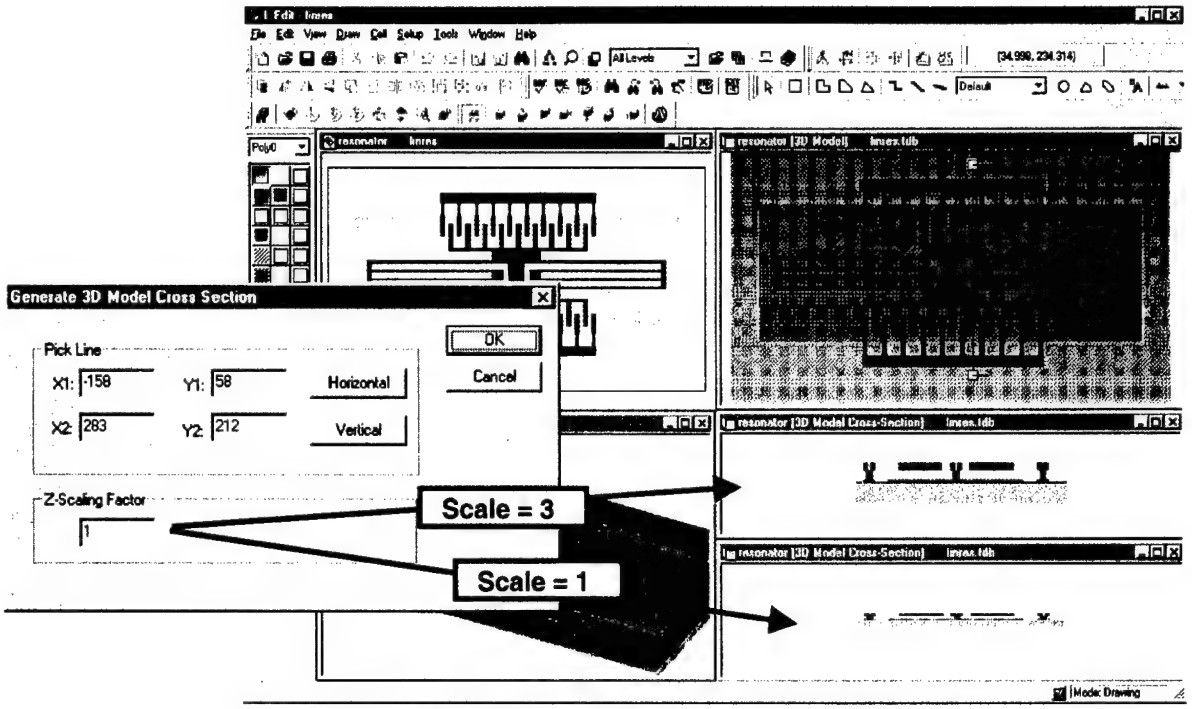


Figure 5: Illustration of height scaling of cross section.

Hide layers in 3D Viewer

In many cases the layer of interest in a device is hidden by other layers or obscured from sight in some way. During this quarter Tanner has implemented the ability to hide any layer in the 3D model, thus bringing the layers under the hidden layer into view. A hidden layer may easily be brought back into view as well.

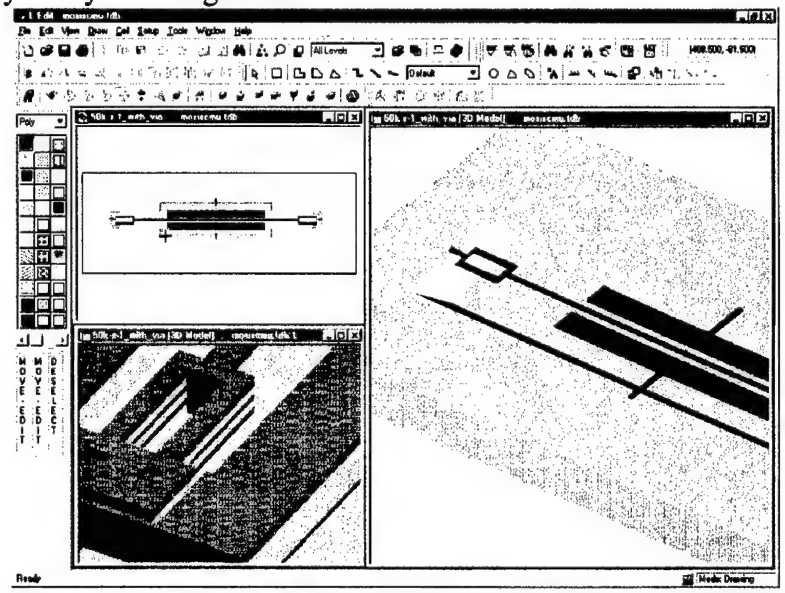


Figure 6: Rome Labs design, illustrating hidden layers feature of 3D modeling tool. Layout (top-left), 3D model close up with all layers (Bottom-left), and 3D model close up with Metal2, Metal3, and oxide hidden (right).

Recent Developments

Additional work this period includes development not included in the release version, but may be released in a future version. Stretching of the vertical axis in the cross section view, export of only the visible portion of the 3D model, as well as conformal deposition onto angled sidewalls was implemented this period.

1.2.2. Process Specification

The process specification is a set of steps that is used to describe a fabrication process. Tanner has developed a process specification that is an abstraction of the true fabrication process, as it describes steps in terms of their geometric *effect*, rather than in terms of material and chemical *inputs*. It provides the Solid Model Generator with instructions on building a 3-D model from the 2-D layout.

Highlights:

- Developed fabrication process definition format adopted by Composite CAD as standard.
- Iterated review of the fabrication process definition format among Composite CAD members.
- Completed a BNF description of the process definition specification.
- Implemented parser to read process definition format.
- Created process definition files for standard processes including MUMPS, SANDIA ITT, and Analog Devices/MCNC iMEMS.

A main goal in the development of the format is that it be easily extensible to future process steps and parameters. In this regard, the most notable feature of the format to come out of the Composite CAD discussion involves combining several similar steps into a single step with a type parameter. This leads to a system of hierarchy and inheritance, and enables easier extension to new types, as well as allows a default behavior to be executed if a particular type is not supported. In particular,

- Conformal Deposit, Snowfall Deposit, and Fill, are combined into a single "Deposit" step, with a DepositType parameter to indicate the type of deposition. This implementation is more extendable to allow for a default behavior, and to add more deposit types.
- The separate steps for Etch, Bulk Etch, and Sacrificial Etch are combined into a single "Etch" step, with an "EtchType" parameter to indicate the type of etch.

Some sample steps from the MUMPS process are shown in Figure 7 below.

```
Wafer={
        MaskName="substrate"
        Thickness=100
        Comment="Wafer"
}
Deposit={
        DepositType=CONFORMAL
        Face=TOP
        LayerName="Nitride"
        Thickness=.6
```

```
        Scf=c
        Comment="Deposit Nitride"
}
Deposit={
        DepositType=CONFORMAL
        Face=TOP
        LayerName="Poly0"
        Thickness=.5
        Scf=c
        Comment="Deposit Poly0"
}
Etch={
        EtchType=SURFACE
        Face=TOP
        MaskName="Poly0"
        EtchMask=OUTSIDE
        Depth=2
        Angle=90
        Undercut=0
        EtchRemoves="Poly0"
        Comment="Etch Poly0"
}
Deposit={
        DepositType=CONFORMAL
        Face=TOP
        LayerName="Ox1"
        Thickness=2
        Scf=.5
        Comment="Deposit Ox1"
}
Etch={
        EtchType=SURFACE
        Face=TOP
        MaskName="Dimple"
        EtchMask=INSIDE
        Depth=.75
        Angle=87
        Undercut=0
        EtchRemoves="Ox1"
        Comment="Etch Dimple"
}
Etch={
        EtchType=SURFACE
        WaferID=w1
        Face=TOP
        MaskName="Anchor1"
        EtchMask=INSIDE
        Depth=2
        Angle=87
        Undercut=0
        EtchRemoves="Ox1"
        Comment="Etch Anchor1"
}
```

Figure 7: Sample Process Definition Steps from MUMPS Process

The process definition may be written as a text file, and parsed by the program, or the user may enter process parameters directly a GUI, as shown in Figure 8. Tanner has also created process definition files for standard processes including MUMPS, SANDIA ITT, and Analog Devices/MCNC iMEMS, which are provided to users with MEMS-Pro.

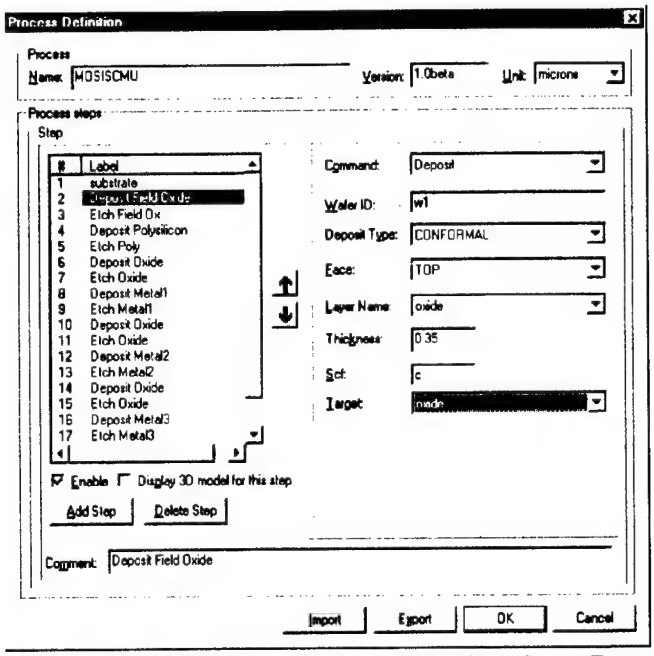


Figure 8: Graphical input option for process definition data. Process may be imported from text file by pressing the "Import" button.

1.3. Libraries and Technology Setup Files

Under this effort Tanner has implemented technology setup files for a number of processes, including MCNC/MUMPS, MOSIS-CMU, Sandia-ITT, and ADiMEMS, and are distributing them in the latest release of MEMS-Pro. The technology files include layer setups, DRC rules, extraction setups, as well as process definition setup for 3D modeling.

In a related effort, Tanner has developed a set of parameterized layout libraries for resonator elements and have developed corresponding schematic and simulation models for each layout element. For each element, a synchronized three view system of layout, schematic, and simulation is provided. Tanner has also developed a graphical interface for easy point-and-click selection of desired components. The graphical interface displays icons of each component grouped by functional category in a tabbed dialog, as shown in Figure 9 below.

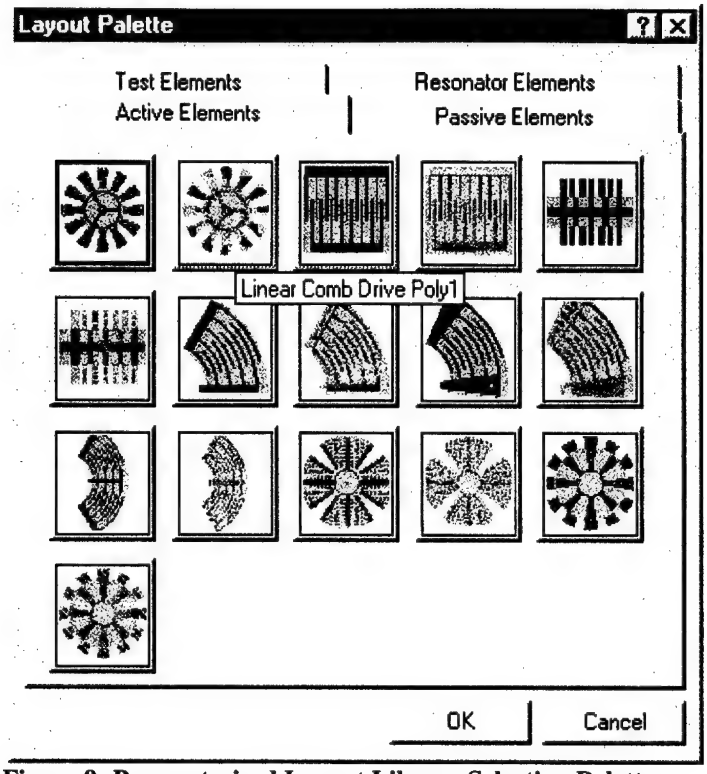


Figure 9: Parameterized Layout Library Selection Palette.

Clicking on the icon for the Comb Drive resonator element, for example, brings up the dialog box shown below in Figure 10. Here the user may enter the parameters for the creation of the Comb Drive, such as width and length of the teeth, width of the gap, and number of gaps.

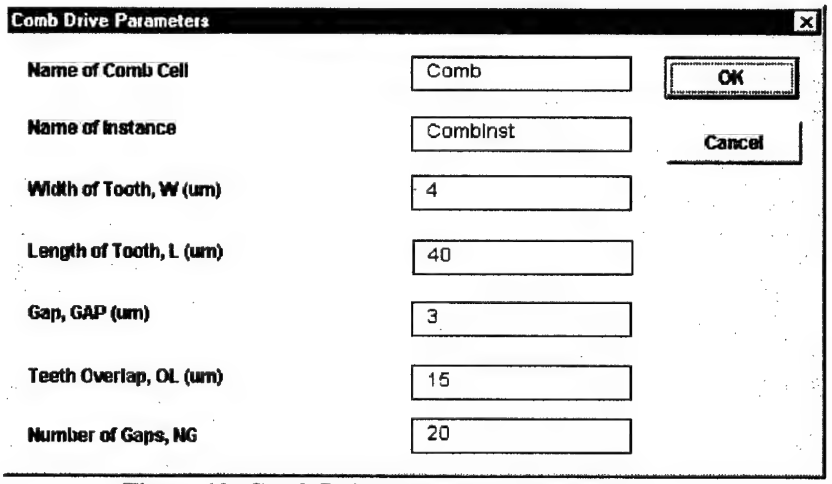


Figure 10: Comb Drive Parameters Dialog Box.

1.3.1. Canonical Design Problem: High Q Resonator

As an illustration of the Library capabilities and concept, Tanner has worked with CMU on the canonical design problem of the High Q Resonator. Tanner has developed

all three views of this device, ie, schematic symbol, simulaton model, and layout macro. Tanner has developed 3D analytical models for the resonator using MEMS Pro to match the CMU results. A schematic of the resonator was created in S-Edit and the models were implemented using S-Edit schematics with links to T-Spice external functional models. Figure 11 below shows the S-Edit schematics, T-Spice netlist, W-Edit simulation results, and L-Edit layout of a particular resonator design.

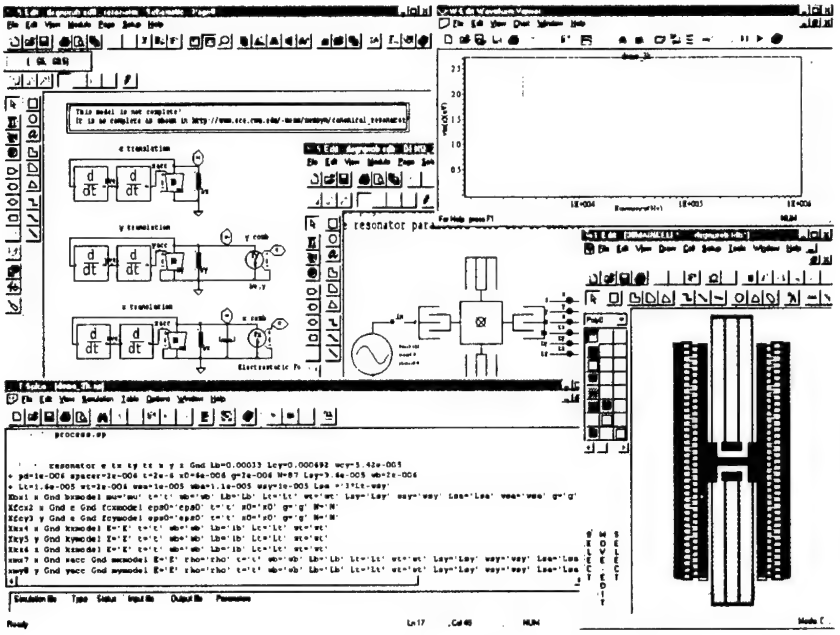


Figure 11. Canonical Design Problem

1.4. Finite/Boundary Element Analysis

This task consists of the development of Self-Consistent Finite Element and/or Boundary Element analysis capability for multidomain Electrostatic, Mechanical, and Thermal analysis. Initial efforts centered on development of solvers either from scratch or based on University developed solvers. Progress from this approach was evaluated midway through the program, and a decision was made that partnering with a commercial FEA company was a better strategy. Tanner established a joint development relationship with ANSYS as well as have developed a strategy for inclusion of the ANSYS tools in MEMS-Pro. The ANSYS software provides capabilities for coupled multiphysics analysis, including mechanical, thermal, electrical, magnetic, and fluidic analysis. The tool is capable of analyzing material nonlinearity, geometric nonlinearity, large deflection analysis, as well as contact nonlinearities. Although 3rd party solvers are being used, Tanner is developing its own mesh generation technology that will be superior to ANSYS mesh generation for MEMS structures.

Tanner's approach follows a two phase approach toward integration. In the first phase, MEMS-Pro provides 3D modeling, meshing, and problem definition (Boundary conditions), and outputs a data file for input into ANSYS. In the second phase, the ANSYS solver is completely embedded in MEMS-Pro, and the user interacts entirely through the MEMS-Pro user interface. Tanner's goal is to deliver a well integrated and

easy to use with ANSYS as the core solver engine. The user thus has the ease of use of the MEMS-Pro integrated design suite, with the power of the ANSYS engines underneath. Additional details on technical advancements toward integration of ANSYS into the Tanner tool suite are provided below.

1.4.1. Mesh Generation

Highlights:

- Developed automatic mesh generation algorithm, tightly interfaced with solid model, and optimized for MEMS/IC fabrication process derived structures.
- Finite element and Boundary element meshing.
- Independent mechanical, electrical, and thermal meshing.
- Support of curved geometry in mesher.
- Continued work on meshing approaches for angled sidewall meshing.

Mesh generation follows solid modeling in the analysis sequence. Tanner has developed a fully automatic meshing capability that requires no user assistance. Automatic mesh generation is especially important, since manual meshing is a laborious process often requiring several days to mesh a complex device. Both finite element volume meshing, and boundary element surface meshing algorithms are developed.

Tanner has developed a finite element meshing approach for that takes maximum advantage of the layered, process derived structure of the device. The algorithm involves projecting the solid model structure onto a 2D plane, meshing the projection, then extruding the 2D mesh into the 3D structure. The planar projection is meshed using a triangulation scheme, and is thus able to accommodate arbitrarily complex configurations. The advantage of this approach, is that the elements may be thin in the z dimension, but very large in the x-y plane, thus reducing the number of elements required to mesh the structure compared to the generalized tetrahedral approach commonly used. Tanner estimates a factor of 10 reduction in the number of elements, compared to tetrahedral meshing. In addition, the Tanner approach covers the domain with the minimum number of triangles required to fit the geometry, subject to constraints of element quality. The approach will then use a measure of the error in the solution to locally refine the mesh to reduce the error to a predefined tolerance. By starting with a coarse mesh, and refining based on the error in the solution, the method can establish the solution to a predefined tolerance, in the minimum required memory.

Finite Element Mesh Generation

The finite element meshing module is tightly interfaced with the solid model module, takes geometry directly from the solid model for meshing. The user simply sets initial mesh parameters, and the code automatically produces a mesh from the 3D model input. An example of a hinge mechanism is shown below. The mesh of 2-D projection of the hinge example is shown in Figure 12, and the corresponding 3-D mesh is shown in Figure 13.

Since the majority of detail in MEMS devices is in the horizontal plane, Tanner has spent effort to ensure high quality and efficient meshing in the triangulation of the solid

model projection. Tanner has investigated several criteria for triangulation refinement, and decided that using a triangle size measure produced superior results over using a triangle shape measure. The algorithm proceeds in roughly the following manner: an initial Delaunay triangulation is given which covers the domain of the computation, the mesh is then iteratively refined by placing nodes in areas which are computed to be "too coarse." The algorithm terminates when no such areas remain on the mesh. A shape criteria (specifically, its aspect ratio, the ratio of the radii of its circumscribing to inscribing circles) provided strict control over the quality of the final mesh by terminating the algorithm when every triangle had aspect ratio equal to or less than a user defined limit. However it frequently produced meshes with too many elements. In an effort to eliminate badly shaped triangles, the algorithm overzealously refined areas of the mesh which could have been more economically covered. Also, because of the post-processing smoothing algorithm used, the quality of the final mesh would very often not satisfy the user limit. A size based criteria terminates when all triangles on the mesh are smaller than their local size function, regardless of their shape. In general, this tends to produce meshes with a minimum number of elements although some of them may be badly shaped. However, optimizations to the point placement strategy and the boundary refinement strategy, and the use of a mesh optimization algorithm have led to an overall algorithm that produces high quality coarse meshes with the fewest reasonably possible number of elements necessary to represent the geometry. Economy of meshes is important, since the goal is to be able to perform analysis on a PC. Performance of the meshing engine has also been optimized, and Tanner has achieved and $O(N \log N)$ complexity of the triangulation algorithm. The algorithm generated tens of thousands of triangles in just a few minutes. Lastly, an element merging postprocessing algorithm was developed to produce a quadrilateral mesh from the triangulation. In the 3D mesh, this results in a higher quality brick mesh vs a tetrahedral mesh.

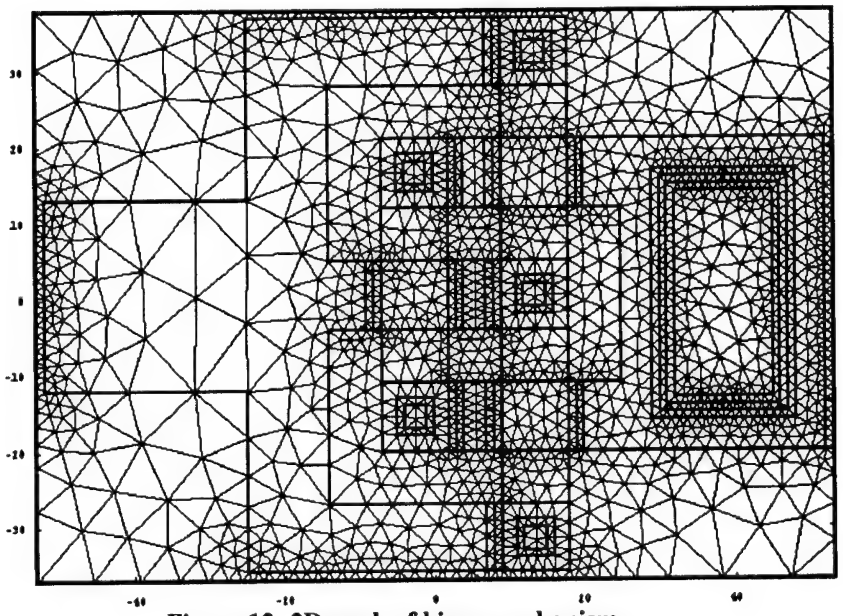


Figure 12: 2D mesh of hinge mechanism.

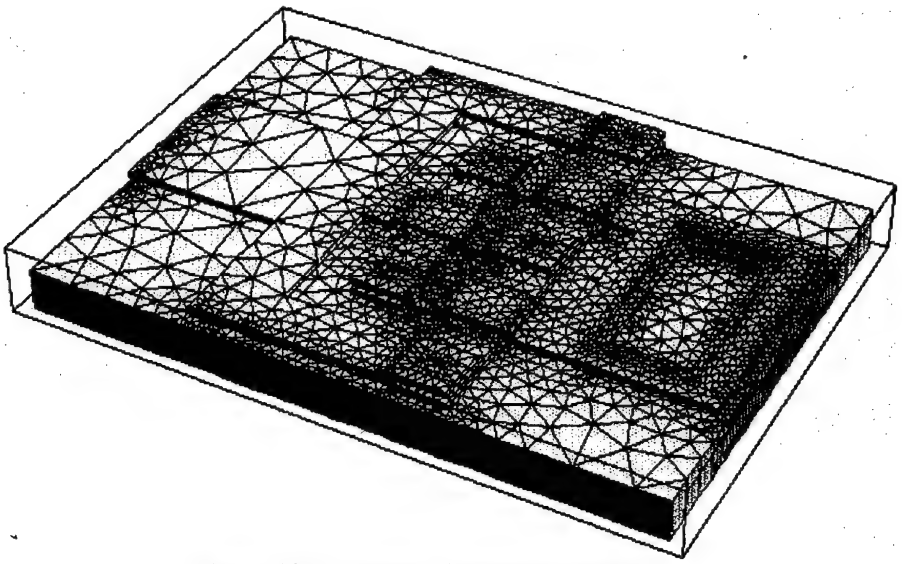


Figure 13: 3-D mesh of hinge mechanism.

Boundary Element Mesh Generation

Tanner's approach toward 3D analysis is to allow a flexible combination of finite element and boundary element solvers to be used. In many cases, a boundary element solver is desired for electrostatic analysis. In this case, only the boundary of the domain need be meshed when using a BE approach, whereas an external unbounded domain must be meshed if a finite element approach is used. Thus, Tanner has developed both boundary element and finite element meshing, and the user will be able to choose the type of solver to be used in a given problem. In the initial approach to BE meshing, the boundary element mesh was simply the surface elements of the finite element mesh. The boundary element meshes produced by this approach were too fine for efficient electrostatic analysis so it was decided that the boundary element and finite element meshes should be completely independent. Boundary element electrostatic solvers typically support nonconforming meshes, so this fact was take advantage of to produce much coarser initial meshes. The mesh is further refined based on an error criteria in the solution, and the solver iterates with mesh refinement until convergence.

Since the electrostatic BE solution needs to be coupled to a structural FE solution for self consistent analysis, an approach has been implemented to perform cross mesh interpolation between the finite element conforming and boundary element nonconforming meshes.

Figure 14 below shows a boundary element mesh of a resonator, obtained by extracting the surface of the finite element structural mesh. Figure 15 shows a nonconforming boundary element mesh of the same resonator, obtained independently of the finite element mesh. Note the dramatic reduction in the number of elements, from 12,585 elements, to 3029 elements, for a 76% reduction. In practice however, FastCap will not be able to obtain a solution on the mesh in Figure 15, due to the high area ratio between the largest and smallest elements. Tests of various structures have shown that

FastCap is able to obtain solution for area ratios of largest to smallest element of about 20:1.

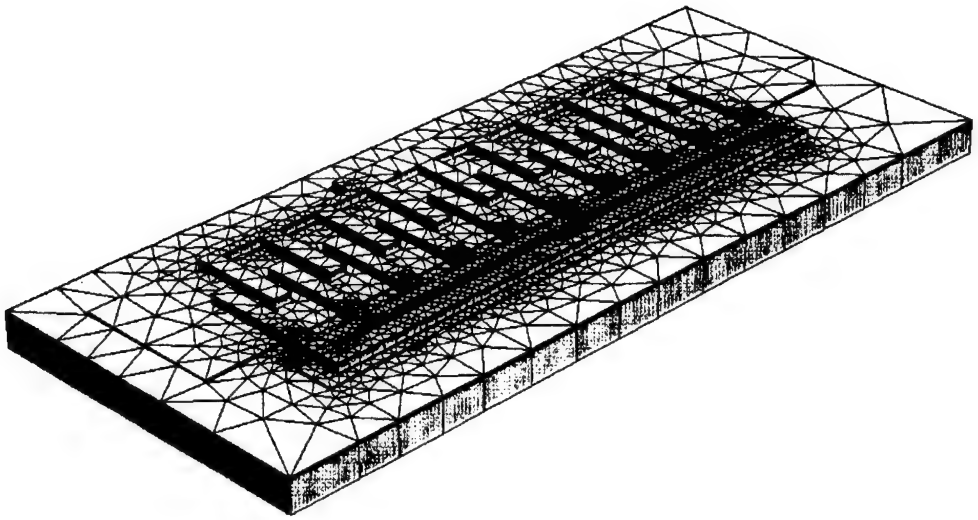


Figure 14: Face extraction method, 12,585 elements.

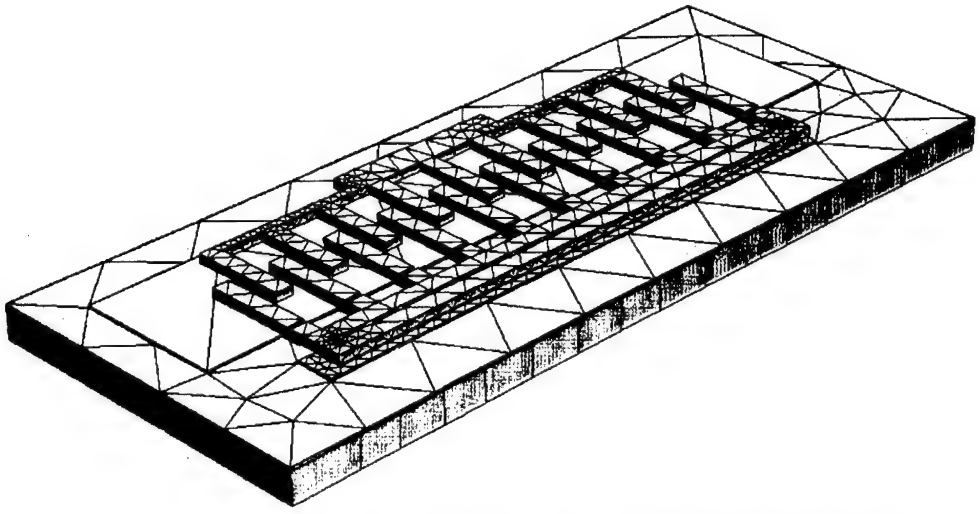


Figure 15: Remeshing algorithm, 3029 elements. (No area ratio constraint)

Applying this constraint to the nonconformal boundary element mesh produces the mesh shown in Figure 16. This mesh contains 9099 elements, for a 28% reduction from the original. While containing fewer elements, the nonconforming mesh in Figure 16 is a

superior mesh for electrostatic analysis as the element distribution is more uniform than the original one obtained by the face extraction method.

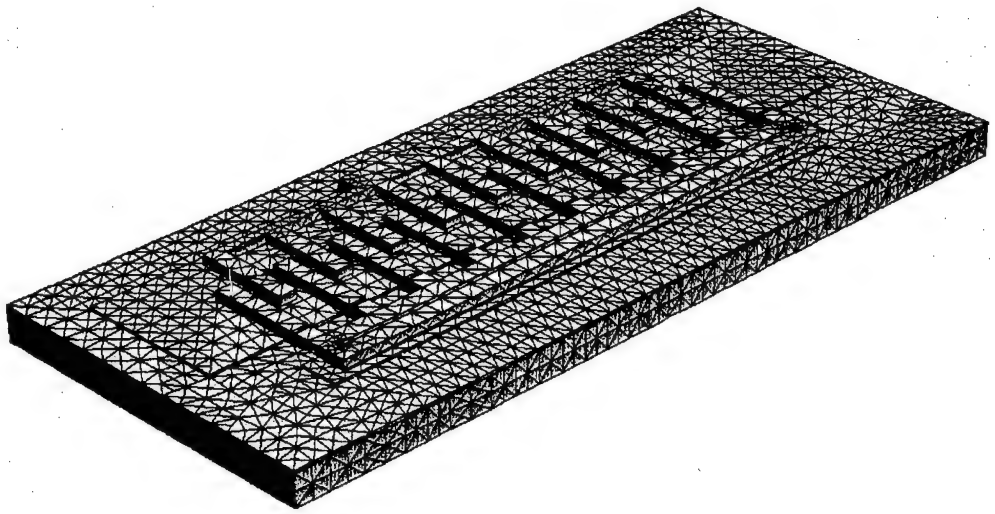


Figure 16:Remeshing algorithm, 9909 elements. (Area ratio limited to 20:1)

ANSYS Link

The finite element mesh generator has the ability to output a mesh in a format directly readable by the ANSYS finite element solver. In addition to the structural FEA mesh, Tanner is able to generate an electrostatic FEA mesh for use by the ANSYS electrostatic solver. Tanner uses the ANSYS "infinite" elements for the far field boundary, and has made needed modifications to the mesher to handle these special elements. The approach essentially consists of building a large sphere of elements around the body, with infinite elements on the outer boundary of the sphere.

Angled Sidewall Meshing

Cases in which only the substrate layer has angled sidewalls can be handled by using a combination of the current Tanner mesher and the ANSYS tetrahedral volume mesher. In this scheme, the Tanner mesher meshes the layers above the substrate. The bottom surface of this mesh are then given to ANSYS along with a solid model of just the substrate layer. ANSYS is then be instructed to generate a volume tetrahedral mesh in this volume while preserving the surface mesh produced by the Tanner mesher. The two meshes are then perfectly matched on the surface defined by the surface mesh.

To handle the general case of angled sidewall solid models Tanner has worked on a new meshing scheme that is a variation on the well-known octree method, but is tailored to produced high quality but very thin elements as the extrusion method does. It will achieve this by allowing the root box of the octree to be a suitably flat rectangular solid and by using bisection rather than octasection to refine the tree (hence it is properly referred to as a bitree method). The bisection algorithm is specified to take into account

the layered nature of the solid model so as to produce as few elements as possible. A preliminary result is shown in the figure below. Note that no layering information was used in this example (only a simple solid model in/out function) so that the bitree is more fine than it otherwise could be.

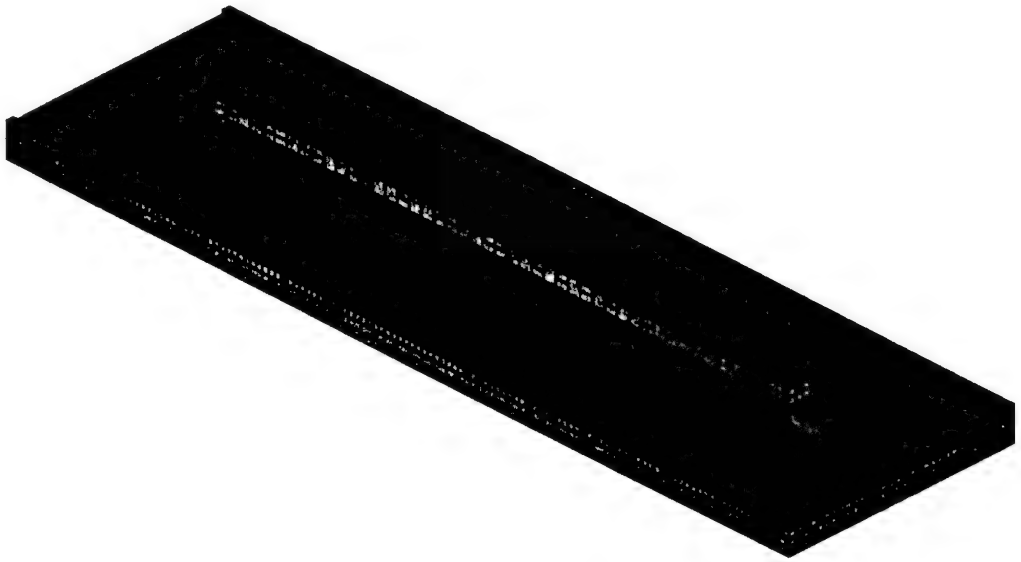


Figure 17: Illustration of new bittree meshing approach.

1.4.2. Finite Element Mechanical, Thermal, and Electrostatic Analysis

Highlights:

- Initial approach was to develop new mechanical and thermal solvers, and use the Fastcap electrostatic solver.
- Strategy changed to interface tightly to the ANSYS FEA solver, with ability to interface to other 3rd party solvers.
- Developed novel approach to boundary conditions, allowing the user to label FEBE boundary conditions on the mask layout.
- Implemented capability to setup capacitance calculation in MEMS-Pro, and invoke ANSYS to calculate the capacitance matrix.

Boundary Conditions/Problem Setup

Tanner has developed an easy to use front end for the finite element analysis tools. By leveraging on its highly integrated layout and 3D modeling environment Tanner has developed a number of capabilities to simplify the analysis process for the designer, particularly in the area of design iteration and boundary condition management. In the typical design environment today, a change in layout requires a new 3D model to be generated for analysis. Boundary conditions on the old model or mesh are lost, and new ones must be created.

The approach is to push as much of the analysis setup and boundary condition information as possible onto the layout. Data specified on the layout will be persistent throughout the design cycle. The user is able to modify the layout and when analysis is invoked, boundary condition data specified on the layout will be automatically transferred to the solid model and then to the mesh. The work of having to reapply boundary conditions every time the design is modified is thus avoided. By using special tags on the layout, the tool is also able to automatically calculate fixed boundary conditions.

Initial Analysis Approach

The initial approach to 3D analysis in this contract was to develop new finite element solvers for mechanical and thermal analysis, and a new boundary element solver for electrostatic analysis. At DARPA's suggestion Tanner modified it'sstrategy for electrostatic analysis to use the MIT developed Fastcap code. Tanner has been successful in developing a prototype structural solver, and coupling the Fastcap solver in a self-consistent solution algorithm. Results from that prototype are shown in Figure 18 depicting the deflection of the sample beam resonator due to the applied voltage.

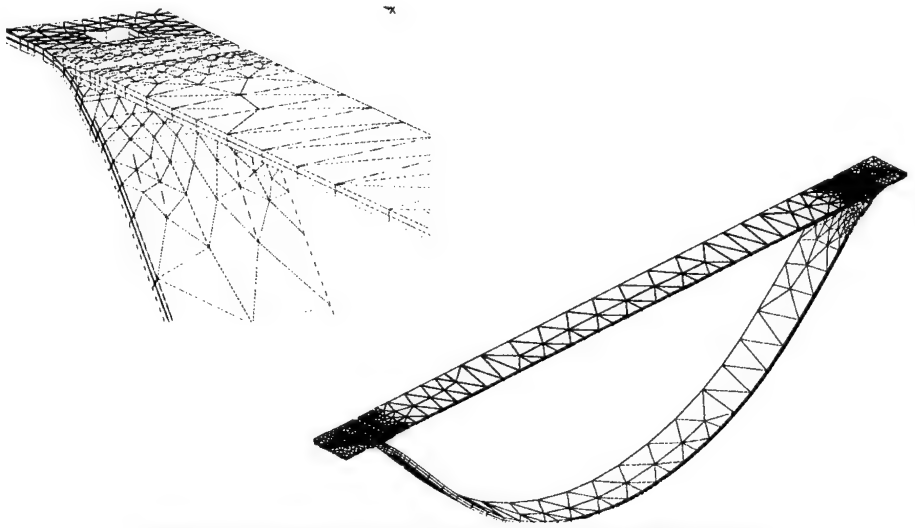


Figure 18: Displacement of beam due to applied voltage.

While this approach enabled a highly customized solver for MEMS applications, several factors led us to conclude that this was ultimatly not the best stategy to continue. Notably, the fastcap solver had difficulty converging on large meshes with large ratios of largest to smallest element size. Also, the Fastcap solver exhibited convergence difficulties when the ratio of the largest to smallest dielectric constant was large. Modifications were made to address these issues, with some improvement, but it appeared that a commercially robust solver would be difficult to achieve. The internally developed structural solver was also successful in prototype runs, but appeared to have serious limitations in capacity. Having identified these shortcomings, the decision was made to interface with the ANSYS finite element solver for 3D analysis. This decision was discussed and approved by Heather Dussult, the DARPA program manager at the time.

ANSYS Interface

Tanner has developed two approaches for easily interfacing from Tanner's MEMS-Pro to ANSYS. In the first approach, the user can write the solid model in a format which can be directly read into ANSYS. The user can then easily import the solid model into ANSYS, and perform any meshing and analysis functions normally available with their ANSYS product. This capability is available in MEMS-Pro V2. Tanner has also developed a second approach, which involves a much more tightly integrated product. In this approach the user meshes the solid model using the Tanner mesher, and controls ANSYS from within MEMS-Pro. This happens in two steps: first MEMS Pro outputs a batch file containing all information necessary for ANSYS to perform a particular analysis, and then instructs ANSYS to execute the commands in the batch file. The batch file manipulations are completely transparent to the user, who controls the process from the MEMS-Pro GUI. Currently the batch file is concluded by instructing ANSYS to write all information about the analysis, including the solution, into a database file which can be used for visualization.

1.4.3. Integrated CoSolver

Highlights for this quarter:

- Developing interface to ANSYS solvers, with ability to setup multiphysics analysis.

Tanner has developed the ability to interface MEMS-Pro to the ANSYS finite element solver for coupled self consistent multidomain analysis. The approach has centered around writing ANSYS command files which instructs the ANSYS engine how to perform the solution in a batch mode operation. Such routines have been written for static structural and electrostatic analysis. The former can also be used to perform a structural modal analysis. Tanner has also prototyped the capability to invoke an ANSYS analysis directly from MEMS-Pro.

Tanner is also performing research into algorithms for coupled multiphysics analysis in order to be able to properly evaluate the ANSYS solvers, and to provide guidance to ANSYS on their approach. In this regard, Tanner is working in collaboration with two members of the applied mathematics department at the California Institute of Technology (Drs. John Pelesko and Patrick Guidotti). In this work a mathematical model of a simple electrostatically actuated MEMS device was constructed and the mathematical properties of the model explored. The purpose is to better understand the nature of this type of coupling with an aim towards improving existing numerical methods for solving the problem. There are essentially two methods in circulation for solving the general 3-D coupled electrostatic-mechanical system: the relaxation method and Aluru and Whites's multi-level Newton method. The former is simpler to implement (it is simply a fixed-point mapping) but suffers from the problem that convergence is slow near the pull-in voltage of a device. The second is significantly more difficult to implement but does not have the convergence problem near pull-in. Tanner has shown that the mathematical model has a "fold" in its solution space and that this fold is what causes the convergence problem in the relaxation method. Several methods developed by Dr. Herbert Keller (also

at Caltech applied mathematics) are known to be effective in dealing with this type of problem and are currently investigating their use in the Tanner model. Tanner expects the results of this investigation to be published in a refereed engineering journal.

1.4.4. Results Rendering

Highlights for this quarter:

- Developed capability for rendering of finite element and boundary element results, and integrated into L-Edit/3D modeling system.

Tanner has developed a finite element/boundary element results prototype rendering module, and integrated this with the L-Edit layout editor and 3D modeling system, bringing full windows functionality to this interface. The rendering supports multiple views, and integration with L-Edit allows the layout, solid model, and results to be managed and viewed all within the single application. Functionality includes panning, zooming, rotation, as well as several postprocessing functions such as calculation and visualization of the three principal stresses (σ_1, σ_2, σ_3), Octahedral Shear Stress and Von Mises Stress. In addition, the user can probe the display with the mouse pointer, and obtain detailed numerical data at the probed location.

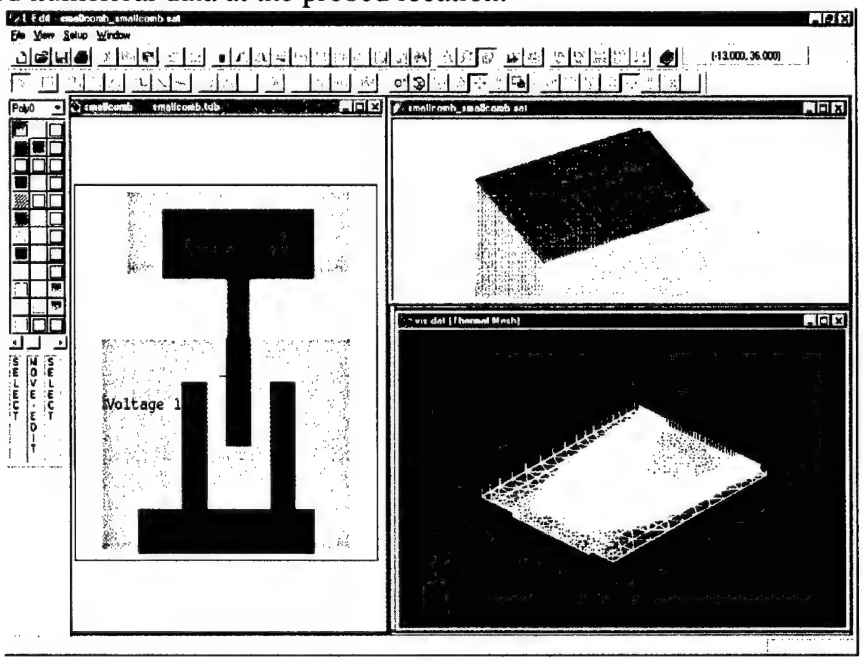


Figure 19: Integrated Layout, Solid Model, and Analysis Results.

1.5. Model Builder

This Task comprises the development of a system that manages and partially automates the development of behavioral models from FE/BE analysis.

Highlights for this quarter:

- Demonstrated one of macromodeling approaches by creating a T-Spice model of a torsional resonator design using ANSYS runs to build tables for the macromodel.
- Held discussions with Selden Crary on the use of the IMSET Design of Experiments program in the macromodeling of an accelerometer.
- Continued implementation of behavioral modeling link to T-Spice.
- Discussed progress, and received progress report from Selden Crary (See Appendix A).
- Continued to refine Model Builder Architecture.
- Continued work on ANSYS link to Model Builder.

1.5.1. Model Builder Architecture

The popular approach to model building so far has been to transform a finite/boundary element based model description to some reduced order model that is suitable for simulation with a differential equation type behavioral simulator, like SPICE, MATLAB or a VHDL-AMS simulator. Tanner feels that the Model Builder architecture should be sufficiently general to encompass reducing/transforming models from a variety of forms. The architecture will support for example, running simulations using a detailed behavioral model and transforming it to a simpler one for a different target behavioral simulator. An example from the VLSI realm is running a SPICE simulation to get a transient analysis waveform and deriving a simpler timing model. For this type of analysis, the model builder cannot make the apriori assumption that it is controlling a suite of finite element tools, but rather must think of controlling a group of simulators, transforming/reducing their output and building an output model. This approach does complicate the design flow, as there are multiple paths through the system, which make the software coding challenging. Although it is not within the scope of this effort to produce a model builder that works with all types of models and simulators, it was important to remove any bias towards having the system only work with finite element simulators. Tanner software support the following input engines- T-Spice, L-Edit/Extract, several FE/BE simulators, and user defined experimental data. In the future, Tanner hopes to support several behavioral input engines (VHDL-AMS type and MATLAB) as well as full-wave solvers. These simulators will be used as Model Builder outputs in the first release.

1.5.2. Model Builder Algorithms

There have been many model building algorithms proposed in the MEMS community, however, there has been no single algorithm that will work for all models, energy domains and simulators. Some algorithms may produce models that are useful for static but not dynamic analysis. So a big part of using these algorithms in Model Builder is to inform the user of their limitations and present each algorithm as appropriate to the user's device. Tanner has taken the approach for the first version of model builder of giving the user a choice of algorithms. Future versions will add more choices.

For the first version several techniques are under consideration. The first is a straightforward, data collection mode leading to a table based model of the device.

Tanner made sure that the proposed model building architecture allowed us to describe the table to be created, implement setting up the various simulators for the analysis, collecting data from them, and generating the output tables. MATLAB implementations will follow. The next section describes a macromodel building example of a torsional resonator. A macromodel of the device for T-Spice using table controlled sources along with subcircuit models is built, and a set of finite element runs is performed using ANSYS to develop the tables for the T-Spice model.

Another approach is to use model reduction techniques of the PVL-Pade Via Lanczos type, as reported in the previous quarter. Tanner has some experience with these types of networks through work on coupling finite element thermal solvers with behavioral electronics simulators. Tanner has prototyped this type of algorithm in the thermal domain. The goal in implementing this algorithm is to put in place the mechanism where the actual reduction code can be modularized, so that as new techniques are available the code can easily be replaced. The first implementation will be a conservative one, where the user is restricted to using it in its region of accuracy and stability.

The third technique is to offer prewritten model building routines. One popular model is the capacitance/force calculation for structures such as comb drives. Another is to characterize the spring constant of various springs. Tanner has prototyped these algorithms and previously published this work in MSM 98.

1.5.3. Model Builder Example -- Generating a T-Spice Macromodel of an Accelerometer Using IMSET and ANSYS

1.5.3.1. Introduction

A T-Spice macromodel of an accelerometer is generated using design of experiments for finite-element analysis (FEA). The IMSET option of the IOPT[1] tool is used to design the optimal experiments to perform ANSYS analysis at to compute the best estimate of model coefficients. This exercise is based on the work of Y. Gianchandani and S. Cray.[2]

The accelerometer is composed of a rigid proof mass supported by four L-shaped beams. The accelerometer's response, the deflection in the z direction, is modeled as a function of two structural and three environmental parameters. These are the point of attachment of the support beam to the proof mass (a), the width of the short segment of the support beam (b), input axis acceleration (a_z), cross axis acceleration (a_x), and temperature (T).

---

[1] S. B. Crary, J. R. Clark, and K. J. Kuether, *IOPT User's Manual*, Center for Integrated Sensors and Circuits Solid-State Electronics Laboratory, The University of Michigan, Ann Arbor, June 1999.

[2] Y. B. Gianchandani and S. B. Crary, "Parametric Modeling of a Microaccelerometer: Comparing I- and D-Optimal Design of Experiments for Finite-Element Analysis", Journal of Microelectromechanical Systems, Vol. 7, No. 2, June 1998

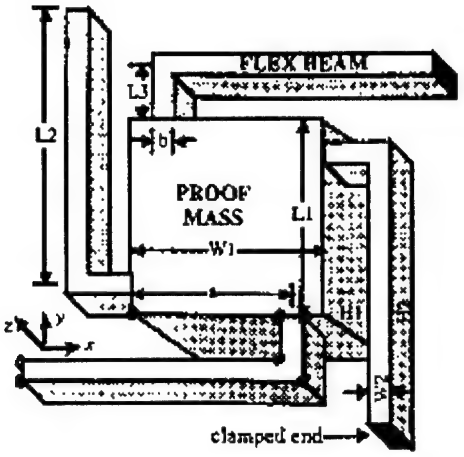


Figure 20. Accelerometer

1.5.3.2. Procedure for Macromodel Generation

The following steps were taken to generate the macromodel:

1. Choose structural variables, environmental variables and ranges, model function
2. Design the experiment by running IMSET
3. Create table of values for layout generator
4. Generate layout of the accelerometers
5. Generate and export 3D models of the accelerometers
6. Run ANSYS analysis and store results
7. Run IMSET analysis to find the model coefficients
8. Create T-Spice model

1.5.3.3. Set Up

All structural dimensions are fixed except two, *a* and *b*. These are listed in Table 1. The design variables and their ranges are listed in Table 2. The response of interest is the displacement in the *z* direction at the center of the top face of the proof mass. It is hypothesized to be a second-order function of *a* and *b* and linear in acceleration and temperature. The model equation is shown in **Equation 1.**

L1 [μm]	1000	Length of the proof mass.
W1 [μm]	1000	Width of the proof mass.
H1 [μm]	500	Height of the proof mass.
L2 [μm]	1000	Length of the long segment of the support beam.
W2 [μm]	50	Width of the long segment of the support beam.
H2 [μm]	10	Height of the support beam.
L3 [μm]	60	Length of the short segment of the support beam

Table 1. Fixed Structural Dimensions and their Values

a [μm]	100 - 900	Structural variable representing the point of attachment of the support beam to the proof mass.
b [μm]	20 - 100	Structural variable representing the width of the short segment of the support beam.

a_z [m/s^2]	0 - 100	Environmental variable representing the acceleration in the input, *z*, direction.
a_x [m/s^2]	1 - 100	Environmental variable representing the acceleration in the cross axis, *x*, direction.
T [°C]	-100 - +100	Environmental variable representing the temperature.

Table 2. Design Variables and their Ranges

$$
\begin{aligned}
Z = {} & \beta_0 + \beta_1 a + \beta_2 b + \beta_3 a_z + \beta_4 a_x + \beta_5 T + \beta_6 a^2 \\
& + \beta_7 b^2 + \beta_8 ab + \beta_9 a a_z + \beta_{10} a a_x + \beta_{11} a T + \beta_{12} b a_z \\
& + \beta_{13} b a_x + \beta_{14} b T + \beta_{15} a_z a_x + \beta_{16} a_z T + \beta_{17} a_x T \\
& + \beta_{18} a^2 a_z + \beta_{19} a^2 a_x + \beta_{20} a^2 T + \beta_{21} b^2 a_z + \beta_{22} b^2 a_x \\
& + \beta_{23} b^2 T + \beta_{24} a b a_z + \beta_{25} a b a_x + \beta_{26} a b T + \beta_{27} a a_z a_x \\
& + \beta_{28} a a_z T + \beta_{29} a a_x T + \beta_{30} b a_z a_x + \beta_{31} b a_z T \\
& + \beta_{32} b a_x T + \beta_{33} T a_z a_x + Z(a,b,a_z,a_x,T)
\end{aligned}
$$

Equation 1. Model Function

1.5.3.4. Designing the Experiments

Since the model function (**Equation 1**) has 34 terms, the minimum number of points required to determine the coefficients is 34. 38 points were chosen to get a better estimate of the coefficients. IMSET was run to generate 38 experiment points for the model function. These points are listed in the table below.

Point Number	A	b	ax	az	T
1	500.000	60.000	84.912	15.088	-69.824
2	180.277	27.463	79.801	79.801	59.602
3	504.971	25.973	83.419	83.419	-66.837
4	819.723	92.537	79.801	20.199	-59.602
5	180.277	27.463	20.199	20.199	59.602
6	495.029	94.027	83.418	16.581	66.837
7	504.972	25.973	16.582	16.581	-66.837
8	495.029	94.027	16.582	16.582	-66.837
9	824.522	27.921	20.184	20.184	59.631
10	824.522	27.921	79.816	20.184	-59.632
11	819.723	92.537	20.199	79.801	-59.602
12	180.277	27.463	20.199	79.801	-59.602
13	839.689	60.491	17.985	17.985	-64.030
14	175.478	92.079	20.184	79.816	-59.631
15	175.478	92.079	20.184	20.184	59.632
16	500.000	60.000	15.088	84.912	-69.824
17	839.689	60.491	82.015	82.015	-64.030
18	175.478	92.079	79.816	20.184	-59.631
19	160.311	59.509	17.985	82.015	64.030
20	495.029	94.027	83.418	83.418	-66.837
21	180.277	27.463	79.801	20.199	-59.602
22	839.689	60.491	17.985	82.015	64.030
23	824.522	27.921	79.816	79.816	59.631
24	839.689	60.491	82.015	17.985	64.030
25	496.188	78.752	50.000	50.000	0.000
26	504.971	25.973	83.419	16.582	66.837
27	819.723	92.537	20.199	20.199	59.602
28	160.311	59.509	17.985	17.985	-64.030
29	160.311	59.510	82.015	82.015	-64.030
30	175.478	92.079	79.816	79.816	59.632
31	503.811	41.248	50.000	50.000	0.000
32	500.000	60.000	15.088	15.088	69.824
33	824.522	27.921	20.184	79.816	-59.632

34	495.029	94.027	16.581	83.419	66.837
35	500.000	60.000	84.912	84.912	69.824
36	819.723	92.537	79.801	79.801	59.602
37	160.311	59.509	82.015	17.985	64.030
38	504.971	25.973	16.582	83.418	66.837

Table 3. Experiment Points

1.5.3.5. Creating the Table of Values for Layout Generation

The geometric parameters (*a* and *b*) required by the layout generator were extracted and saved as a text file in a format that is recognized by the layout generator macro. The contents of the data file (*accel.dat*) is shown below.

a (um)	b (um)	Cell Name
500	60	N1_A500_B60
180	27	N2_A180_B27
505	26	N3_A505_B26
820	93	N4_A820_B93
180	27	N5_A180_B27
495	94	N6_A495_B94
505	26	N7_A505_B26
495	94	N8_A495_B94
825	28	N9_A825_B28
825	28	N10_A825_B28
820	93	N11_A820_B93
180	27	N12_A180_B27
840	60	N13_A840_B60
175	92	N14_A175_B92
175	92	N15_A175_B92
500	60	N16_A500_B60
840	60	N17_A840_B60
175	92	N18_A175_B92
160	60	N19_A160_B60
495	94	N20_A495_B94
180	27	N21_A180_B27
840	60	N22_A840_B60
825	28	N23_A825_B28
840	60	N24_A840_B60
496	79	N25_A496_B79
505	26	N26_A505_B26
820	93	N27_A820_B93
160	60	N28_A160_B60
160	60	N29_A160_B60
175	92	N30_A175_B92
504	41	N31_A504_B41
500	60	N32_A500_B60
825	28	N33_A825_B28
495	94	N34_A495_B94
500	60	N35_A500_B60
820	93	N36_A820_B93
160	60	N37_A160_B60
505	26	N38_A505_B26

Table 4. Parameter Values for Layout Generator

1.5.3.6. Generating the Accelerometer Layout

The layout generator for the accelerometer is an L-Edit/UPI macro. This section describes how to load the macro into L-Edit, specify the input parameters, and launch the layout generation.

Loading the Macro

- ☑ Launch L-Edit.
- ☑ Choose **File > New** from the menu bar.
- ☑ Choose **Tools > Macro** from the menu bar.
- ☑ Click **Add** and open the file *accel.dll.*

Running the Macro

- ☑ Choose **Tools > Generate Accelerometer**. The **Accelerometer Parameters** dialog will appear.

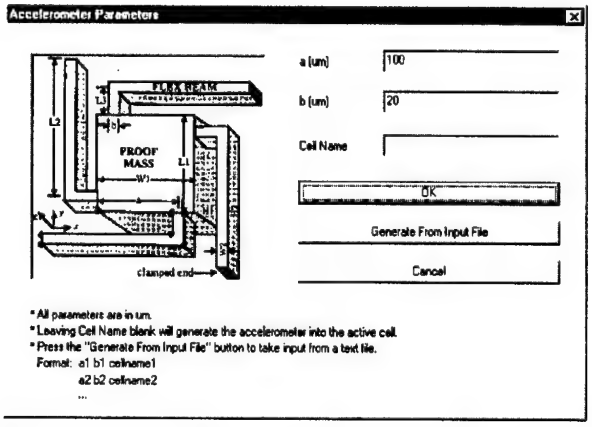


Figure 21. Accelerometer Parameters Dialog

User Interface

There are two ways of launching layout generation. The first is by entering the parameters into the edit fields and then clicking the **OK** button. The second is by clicking the **Generate From Input File** button and then choosing the file containing the list of parameters.

Parameters

a (μm)	100	Location of attachment of the FLEX BEAM.
b (μm)	20	Width of the attachment of the FLEX BEAM to the PROOF MASS.
Cell Name		Name of cell to generate the layout into. If this field is filled in, a new cell is created and the accelerometer is generated into that new cell. If this field is left blank, the layout is generated into the current active cell. If this cell exists, layout generation

		is aborted.

Controls

OK	Launches layout generation using the input values in the edit fields for **a**, **b**, and **Cell Name**.
Generate From Input File	Requests input file. The format of the file is described in the next section on **Input File Format**.
Cancel	Cancel layout generation.

Input File Format

The format of the input file is described here. The input files are ASCII text files containing columns of values for **a**, **b**, and **Cell Name**. Each set of input parameter values is separated by a new line character and each parameter value is separated by white space (spaces, tabs). For example:

```
500   60    N1_A500_B60
180   27    N2_A180_B27
505   26    N3_A505_B26
...
```

Tutorial

Getting Started

- ☑ Launch L-Edit.
- ☑ Choose **File > New** from the menu bar.
- ☑ Choose **Tools > Macro** from the menu bar.
- ☑ Click **Add** and open the file *accel.dll*.

Generating Based on Parameter Values Entered into the Edit Fields

- ☑ Choose **Tools > Generate Accelerometer**. The **Accelerometer Parameters** dialog will appear.
- ☑ Accept the default values for **a**, **b**, and **Cell Name** and click **OK** to generate the layout.

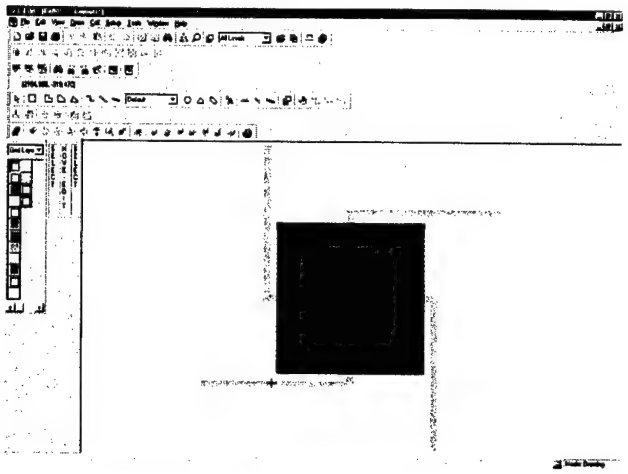


Figure 22. Generated Layout for Default Values of *a* and *b*

Generating Based on Parameter Values Input File

☑ Choose **Tools > Generate Accelerometer**. The **Accelerometer Parameters** dialog will appear.

☑ Click **Generate From Input File** and choose the file *demo.dat*. This file contains the following information:

```
500   60    N1_A500_B60
180   27    N2_A180_B27
505   26    N3_A505_B26
820   93    N4_A820_B93
180   27    N5_A180_B27
495   94    N6_A495_B94
```

Each layout will open in its specified cell. The following is a tiled view of the generated layout.

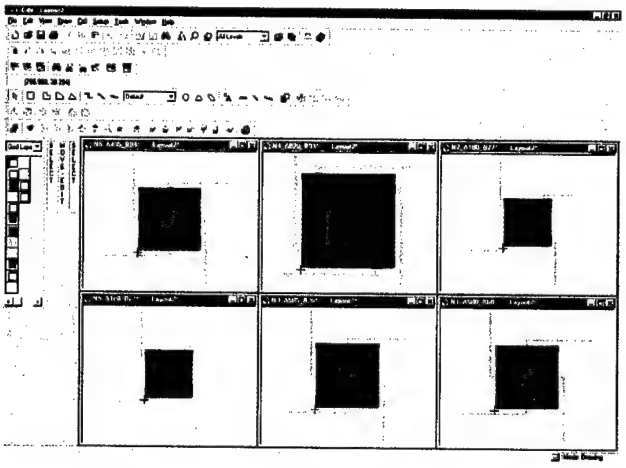


Figure 23. Generated Layout using *demo.dat*

1.5.3.7. Layout Generated for IMSET Values

For this exercise, 38 accelerometers are generated using the parameter values generated by IMSET. The parameter values are listed in a table in Table 4 and the file *accel.dat*. The generated accelerometers are shown in the figure below and in the file *accel.tdb*:

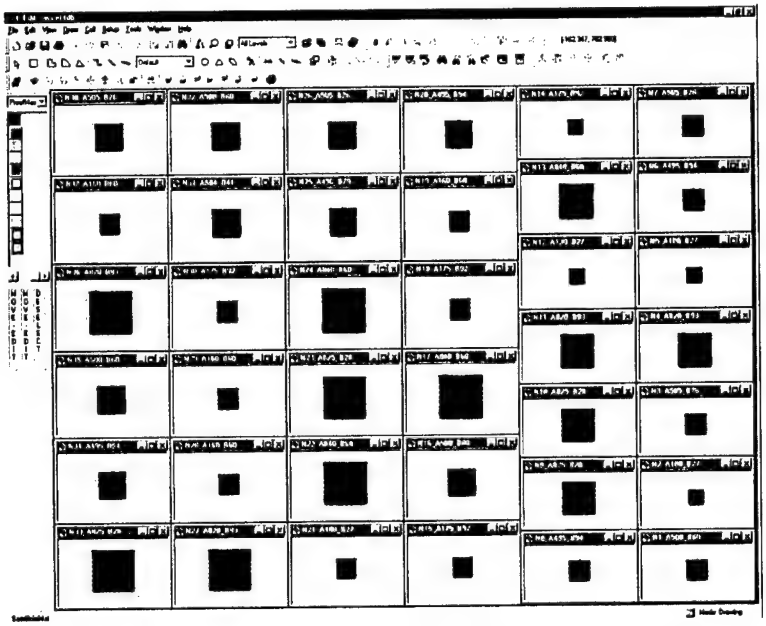


Figure 24. Generated Layout using *accel.dat*

1.5.3.8. Generating and Exporting 3D Models from Layout

For each accelerometer, a 3D Model was generated and exported.

☑ With a layout window active, choose **Tools > 3D Tools > View 3D Model**. The 3D Model of the active cell will be created and displayed.

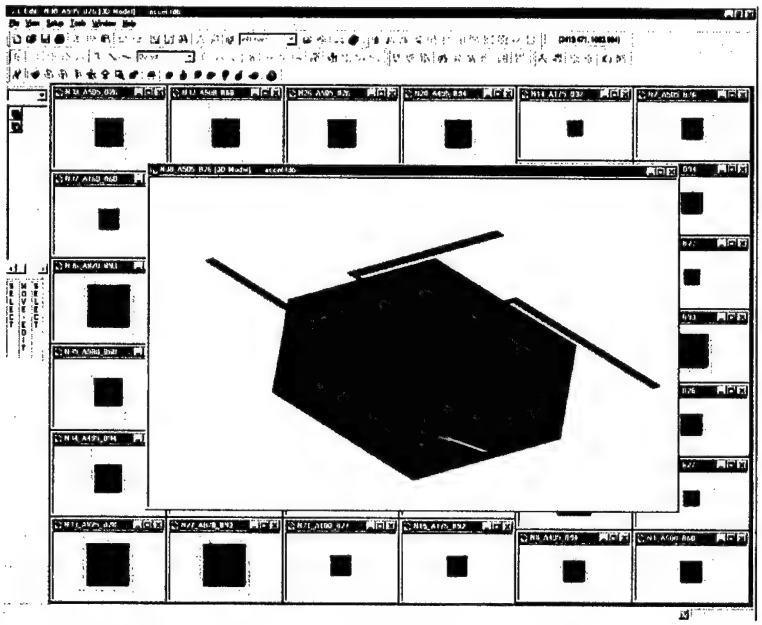


Figure 25. 3D Model Generated for *a*=505, *b*=26, and *Cell Name*="N38_A505_B26"

☑ With a 3D Model window active, choose **Tools > Export 3D Model**. Set the **File name** and click **Export**.

1.5.3.9. Running ANSYS and Storing Results

For each experiment, an ANSYS structural analysis was performed to discover the z displacement of the center of the top face of the proof mass. These results are tabulated below.

Point Number	A	b	ax	az	T	Z
1	500.000	60.000	84.912	15.088	-69.824	-0.4223843
2	180.277	27.463	79.801	79.801	59.602	-2.547131
3	504.971	25.973	83.419	83.419	-66.837	-2.684212
4	819.723	92.537	79.801	20.199	-59.602	-0.5045537
5	180.277	27.463	20.199	20.199	59.602	-0.6447884
6	495.029	94.027	83.418	16.581	66.837	-0.4121259
7	504.972	25.973	16.582	16.581	-66.837	-0.5335132
8	495.029	94.027	16.582	16.582	-66.837	-0.4120998
9	824.522	27.921	20.184	20.184	59.631	-0.6427243
10	824.522	27.921	79.816	20.184	-59.632	-0.6426650
11	819.723	92.537	20.199	79.801	-59.602	-1.993372
12	180.277	27.463	20.199	79.801	-59.602	-2.546955
13	839.689	60.491	17.985	17.985	-64.030	-0.5027006
14	175.478	92.079	20.184	79.816	-59.631	-1.996822
15	175.478	92.079	20.184	20.184	59.632	-0.5050112
16	500.000	60.000	15.088	84.912	-69.824	-2.377215
17	839.689	60.491	82.015	82.015	-64.030	-2.292463
18	175.478	92.079	79.816	20.184	-59.631	-0.5049537
19	160.311	59.509	17.985	82.015	64.030	-2.300195
20	495.029	94.027	83.418	83.418	-66.837	-2.073253
21	180.277	27.463	79.801	20.199	-59.602	-0.6446122
22	839.689	60.491	17.985	82.015	64.030	-2.292494
23	824.522	27.921	79.816	79.816	59.631	-2.541435

24	839.689	60.491	82.015	17.985	64.030	-0.5027308
25	496.188	78.752	50.000	50.000	0.000	-1.311132
26	504.971	25.973	83.419	16.582	66.837	-0.5336149
27	819.723	92.537	20.199	20.199	59.602	-0.5045612
28	160.311	59.509	17.985	17.985	-64.030	-0.5043388
29	160.311	59.510	82.015	82.015	-64.030	-2.30082
30	175.478	92.079	79.816	79.816	59.632	-1.996879
31	503.811	41.248	50.000	50.000	0.000	-1.500716
32	500.000	60.000	15.088	15.088	69.824	-0.4224393
33	824.522	27.921	20.184	79.816	-59.632	-2.541376
34	495.029	94.027	16.581	83.419	66.837	-2.073278
35	500.000	60.000	84.912	84.912	69.824	-2.377270
36	819.723	92.537	79.801	79.801	59.602	-1.993379
37	160.311	59.509	82.015	17.985	64.030	-0.5044521
38	504.971	25.973	16.582	83.418	66.837	-2.684314

Table 5. ANSYS Results for the Experiments

1.5.3.10. Running IMSET Analysis to Find Model Coefficients

Coeff. β_i	Term	IMSET fit for β_i	Coeff. γ_i	Term	IMSET fit for γ_i
β_0	-		γ_1	$Exp(-(a-a^{(1)})^2-(b-b^{(1)})^2-(a_z-a_z^{(1)})^2-(a_x-a_x^{(1)})^2-(T-T^{(1)})^2)$	
β_1	a		γ_2	$Exp(-(a-a^{(2)})^2-(b-b^{(2)})^2-(a_z-a_z^{(2)})^2-(a_x-a_x^{(2)})^2-(T-T^{(2)})^2)$	
β_2	b		γ_3	$Exp(-(a-a^{(3)})^2-(b-b^{(3)})^2-(a_z-a_z^{(3)})^2-(a_x-a_x^{(3)})^2-(T-T^{(3)})^2)$	
β_3	a_z		γ_4	$Exp(-(a-a^{(4)})^2-(b-b^{(4)})^2-(a_z-a_z^{(4)})^2-(a_x-a_x^{(4)})^2-(T-T^{(4)})^2)$	
β_4	a_x		γ_5	$Exp(-(a-a^{(5)})^2-(b-b^{(5)})^2-(a_z-a_z^{(5)})^2-(a_x-a_x^{(5)})^2-(T-T^{(5)})^2)$	
β_5	T		γ_6	$Exp(-(a-a^{(6)})^2-(b-b^{(6)})^2-(a_z-a_z^{(6)})^2-(a_x-a_x^{(6)})^2-(T-T^{(1)})^2)$	
β_6	a^2		γ_7	$Exp(-(a-a^{(7)})^2-(b-b^{(7)})^2-(a_z-a_z^{(7)})^2-(a_x-a_x^{(7)})^2-(T-T^{(1)})^2)$	
β_7	b^2		γ_8	$Exp(-(a-a^{(8)})^2-(b-b^{(8)})^2-(a_z-a_z^{(8)})^2-(a_x-a_x^{(8)})^2-(T-T^{(1)})^2)$	
β_8	a b		γ_9	$Exp(-(a-a^{(9)})^2-(b-b^{(9)})^2-(a_z-a_z^{(9)})^2-(a_x-a_x^{(9)})^2-(T-T^{(1)})^2)$	
β_9	$a\ a_z$		γ_{10}	$Exp(-(a-a^{(10)})^2-(b-b^{(10)})^2-(a_z-a_z^{(10)})^2-(a_x-a_x^{(10)})^2-(T-T^{(1)})^2)$	
β_{10}	$a\ a_x$		γ_{11}	$Exp(-(a-a^{(11)})^2-(b-b^{(11)})^2-(a_z-a_z^{(11)})^2-(a_x-a_x^{(11)})^2-(T-T^{(1)})^2)$	
β_{11}	a T		γ_{12}	$Exp(-(a-a^{(12)})^2-(b-b^{(12)})^2-(a_z-a_z^{(12)})^2-(a_x-a_x^{(12)})^2-(T-T^{(1)})^2)$	
β_{12}	$b\ a_z$		γ_{13}	$Exp(-(a-a^{(13)})^2-(b-b^{(13)})^2-(a_z-a_z^{(13)})^2-(a_x-a_x^{(13)})^2-(T-T^{(1)})^2)$	
β_{13}	$b\ a_x$		γ_{14}	$Exp(-(a-a^{(14)})^2-(b-b^{(14)})^2-(a_z-a_z^{(14)})^2-(a_x-a_x^{(14)})^2-(T-T^{(1)})^2)$	
β_{14}	b T		γ_{15}	$Exp(-(a-a^{(15)})^2-(b-b^{(15)})^2-(a_z-a_z^{(15)})^2-(a_x-a_x^{(15)})^2-(T-T^{(1)})^2)$	
β_{15}	$a_z\ a_x$		γ_{16}	$Exp(-(a-a^{(16)})^2-(b-b^{(16)})^2-(a_z-a_z^{(16)})^2-(a_x-a_x^{(16)})^2-(T-T^{(1)})^2)$	
β_{16}	a_z T		γ_{17}	$Exp(-(a-a^{(17)})^2-(b-b^{(17)})^2-(a_z-a_z^{(17)})^2-(a_x-a_x^{(17)})^2-(T-T^{(1)})^2)$	
β_{17}	a_x T		γ_{18}	$Exp(-(a-a^{(18)})^2-(b-b^{(18)})^2-(a_z-a_z^{(18)})^2-(a_x-a_x^{(18)})^2-(T-T^{(1)})^2)$	
β_{18}	$a^2\ a_z$		γ_{19}	$Exp(-(a-a^{(19)})^2-(b-b^{(19)})^2-(a_z-a_z^{(19)})^2-(a_x-a_x^{(19)})^2-(T-T^{(1)})^2)$	
β_{19}	$a^2\ a_x$		γ_{20}	$Exp(-(a-a^{(20)})^2-(b-b^{(20)})^2-(a_z-a_z^{(20)})^2-(a_x-a_x^{(20)})^2-(T-T^{(1)})^2)$	
β_{20}	a^2 T		γ_{21}	$Exp(-(a-a^{(21)})^2-(b-b^{(21)})^2-(a_z-a_z^{(21)})^2-(a_x-a_x^{(21)})^2-(T-T^{(1)})^2)$	
β_{21}	$b^2\ a_z$		γ_{22}	$Exp(-(a-a^{(22)})^2-(b-b^{(22)})^2-(a_z-a_z^{(22)})^2-(a_x-a_x^{(22)})^2-(T-T^{(1)})^2)$	
β_{22}	$b^2\ a_z$		γ_{23}	$Exp(-(a-a^{(23)})^2-(b-b^{(23)})^2-(a_z-a_z^{(23)})^2-(a_x-a_x^{(23)})^2-(T-T^{(1)})^2)$	
β_{23}	b^2 T		γ_{24}	$Exp(-(a-a^{(24)})^2-(b-b^{(24)})^2-(a_z-a_z^{(24)})^2-(a_x-a_x^{(24)})^2-(T-T^{(1)})^2)$	
β_{24}	$a\ b\ a_z$		γ_{25}	$Exp(-(a-a^{(25)})^2-(b-b^{(25)})^2-(a_z-a_z^{(25)})^2-(a_x-a_x^{(25)})^2-(T-T^{(1)})^2)$	
β_{25}	$a\ b\ a_x$		γ_{26}	$Exp(-(a-a^{(26)})^2-(b-b^{(26)})^2-(a_z-a_z^{(26)})^2-(a_x-a_x^{(26)})^2-(T-T^{(1)})^2)$	
β_{26}	a b T		γ_{27}	$Exp(-(a-a^{(27)})^2-(b-b^{(27)})^2-(a_z-a_z^{(27)})^2-(a_x-a_x^{(27)})^2-(T-T^{(1)})^2)$	
β_{27}	$a\ a_z\ a_x$		γ_{28}	$Exp(-(a-a^{(28)})^2-(b-b^{(28)})^2-(a_z-a_z^{(28)})^2-(a_x-a_x^{(28)})^2-(T-T^{(1)})^2)$	
β_{28}	$a\ a_z$ T		γ_{29}	$Exp(-(a-a^{(29)})^2-(b-b^{(29)})^2-(a_z-a_z^{(29)})^2-(a_x-a_x^{(29)})^2-(T-T^{(1)})^2)$	
β_{29}	$a\ a_x$ T		γ_{30}	$Exp(-(a-a^{(30)})^2-(b-b^{(30)})^2-(a_z-a_z^{(30)})^2-(a_x-a_x^{(30)})^2-(T-T^{(1)})^2)$	

β_{30}	$b\ a_z\ a_x$	γ_{31}	$Exp(-(a-a^{(31)})^2-(b-b^{(31)})^2-(a_z-a_z^{(31)})^2-(a_x-a_x^{(31)})^2-(T-T^{(1)})^2)$
β_{31}	$b\ a_z\ T$	γ_{32}	$Exp(-(a-a^{(32)})^2-(b-b^{(32)})^2-(a_z-a_z^{(32)})^2-(a_x-a_x^{(32)})^2-(T-T^{(1)})^2)$
β_{32}	$b\ a_x\ T$	γ_{33}	$Exp(-(a-a^{(33)})^2-(b-b^{(33)})^2-(a_z-a_z^{(33)})^2-(a_x-a_x^{(33)})^2-(T-T^{(1)})^2)$
β_{33}	$T\ a_z\ a_x$	γ_{34}	$Exp(-(a-a^{(34)})^2-(b-b^{(34)})^2-(a_z-a_z^{(34)})^2-(a_x-a_x^{(34)})^2-(T-T^{(1)})^2)$
		γ_{35}	$Exp(-(a-a^{(35)})^2-(b-b^{(35)})^2-(a_z-a_z^{(35)})^2-(a_x-a_x^{(35)})^2-(T-T^{(1)})^2)$
		γ_{36}	$Exp(-(a-a^{(36)})^2-(b-b^{(36)})^2-(a_z-a_z^{(36)})^2-(a_x-a_x^{(36)})^2-(T-T^{(1)})^2)$
		γ_{37}	$Exp(-(a-a^{(37)})^2-(b-b^{(37)})^2-(a_z-a_z^{(37)})^2-(a_x-a_x^{(37)})^2-(T-T^{(1)})^2)$
		γ_{38}	$Exp(-(a-a^{(38)})^2-(b-b^{(38)})^2-(a_z-a_z^{(38)})^2-(a_x-a_x^{(38)})^2-(T-T^{(1)})^2)$

Table 6. Coefficients for Model Equation

1.5.3.11. Creating the Macromodel

The macromodel for the accelerometer was implemented using S-Edit as the schematic front end. Shown below in

Figure **26** is the *Symbol* view of the accelerometer element. The structural parameters *a* and *b* and the coefficients of the model *β0 - β33* are properties of the symbol. The environmental parameters a_z, a_x, and *T* are ports of the symbol. The voltages at these ports represent the accelerations in the *z* and *x* directions and the temperature. The *z* displacement is modeled using an expression voltage controlled voltage source (the E element).

A very simple circuit was constructed to test this macromodel. The test circuit shown in Figure 27 performs a DC sweep of the acceleration in the *z* direction, a_z. As expected, the plot of *Z* vs. *az* is linear.

Note that this macromodel *only captures the mechanical behavior* of the accelerometer. In a microaccelerometer *system*, the *z* displacement may be detected in a number of ways. If the detection method is capacitive, a *z* displacement controlled capacitor should be added to the macromodel. If the detection method is piezoresistive, a variable resistor should be added to the macromodel. etc. Measurement circuits must also be added to detect these values.

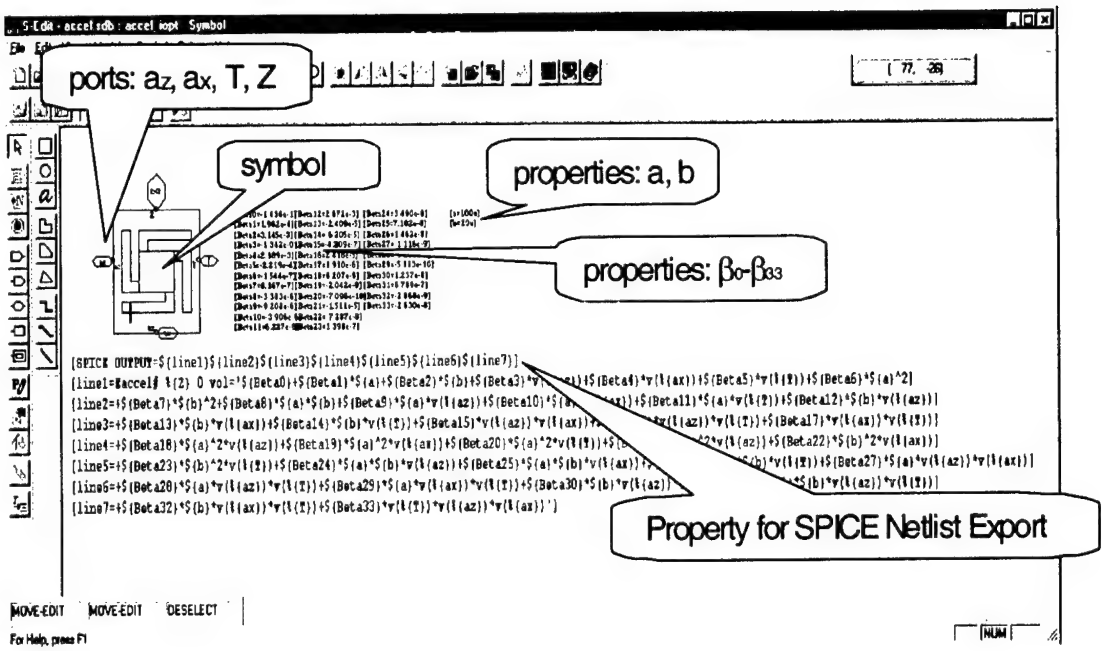


Figure 26. Schematic Symbol of the Accelerometer Macromodel

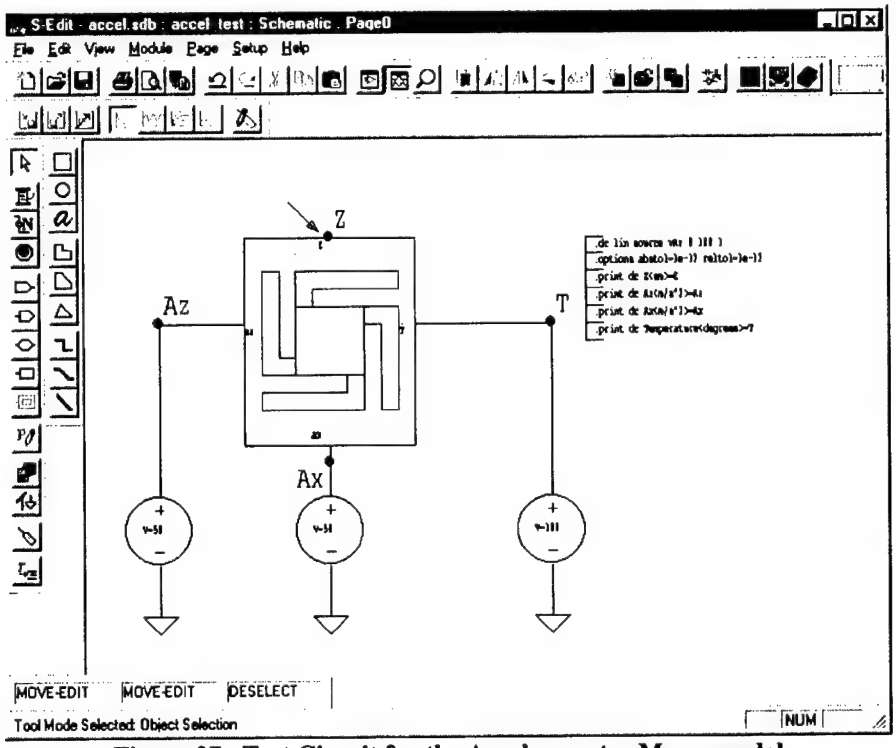


Figure 27. Test Circuit for the Accelerometer Macromodel

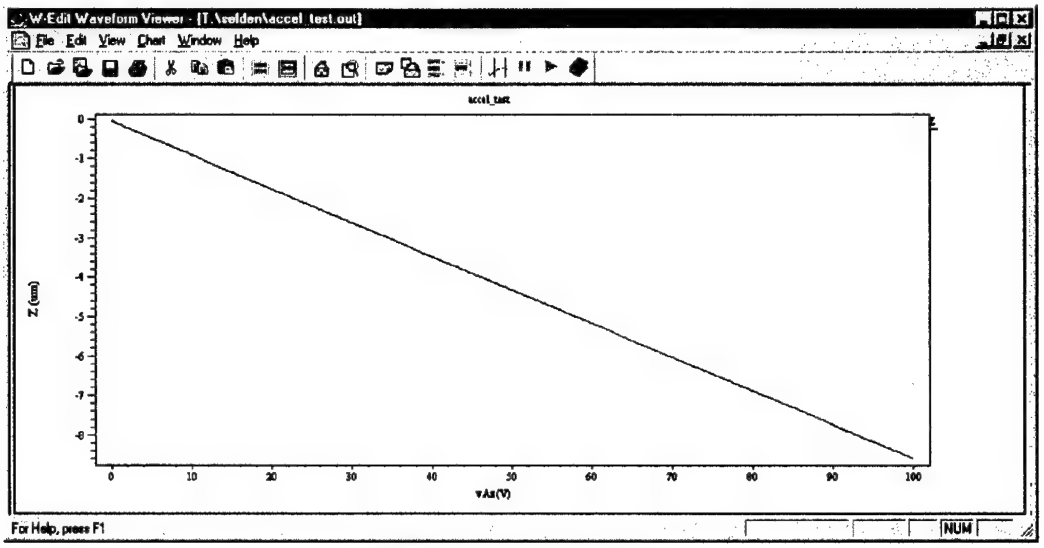


Figure 28. Plot of Z vs. A_z

1.5.3.12. Future Work

The macromodel will be extended to include the electrical behavior as a function of the z displacement. The electrical behavior will be coupled to the mechanical model implemented in the section above through the Z port. Detection circuits will also be added to exercise this macromodel in an accelerometer system.

Each step along the path from device structure to macromodel generation was demonstrated here. While the instructions are simple, many of the steps were performed manually. Most of these steps will be automated to simplify and shorten the design cycle. Future work will include better integration and smoother flow of data between layout, picking of the factors and model equation, design of experiment, 3D model generation and export, finite element analysis, results gathering, calculating the model coefficients, and actual building of the macromodel.

1.6. Circuit Refinement Suite and Mixed-Language Macromodel Support

The tasks under this effort are directed at creating a system level simulation and analysis capability to complement the device level layout and analysis tools currently under development. Although an HDL-only approach is preferable for all-digital circuits, design entry for mixed-signal (analog/digital/RF) electronic circuits is best accomplished using a combined schematic/HDL approach. The same is true for mixed-technology microchips, where IM components are mixed with standard electronic circuits. The ability to perform design entry and simulation using a blend of components from various domains (electrical, mechanical, optical, fluidic, etc.) motivates the schematic, behavioral modeling, and simulation capabilities of this section. Tasks in this section include i) Schematic Tool Multi-domain IM Enhancements, ii) Thermal & Mechanical Behavioral Modeling Engines, iii) Mixed-Language Models, and iv) MATLAB Modeling capability.

Highlights:

- Integration of Viewlogic's Viewdraw schematic capture tool into Tanner's design and simulation environment.
- Implemented a "C-Code" functional model interface to T-Spice.
- Implemented MATLAB model interface to T-Spice.
- Demonstrated ability to perform mixed language model simulations involving Spice, "C", Data-Table, and MATLAB models.

1.6.1. Schematic Tool Multi-domain Enhancements

The goal of this task is to create a schematic capability with domain dependent typed signals with error checking, mixed domain busses, and multi-domain symbols display. This gives the Tanner design environment front-end tools the ability to manipulate mixed-domain schematics, with various signal types, and view simulated mixed-domain signals. The schematic editor will also be enhanced with support for test benches for controlling simulation that have domain specific stimuli and control parameters. Input forces, pressures or acceleration, tolerance settings for spatial variables are examples of stimuli requiring domain specific treatment.

Tanner's approach to this task is to integrate the ViewDraw schematic editor into the MEMS-Pro tools suite, then to develop additional capabilities through the ViewDraw API. The ViewDraw schematic editor is able to handle very large designs, and many capabilities of interest to mixed technology designers are already present in the software, including busses and multidomain model manipulation. In addition, ViewDraw has a flexible API that allows us to implement additional MEMS features within the tool. The integration is proceeding in two phases, the first release provides file level communication between ViewDraw and T-Spice, component libraries in the ViewDraw format, and a translation capability to export designs from S-Edit into ViewDraw. The second phase implements a tight integration between ViewDraw, T-Spice, and W-Edit, providing the ability to invoke simulations from ViewDraw, and perform cross-probing of waveforms by clicking on nodes in the schematic.

Tanner has also worked on support of busses and complex numbers in T-Spice and W-Edit, and have investigated how busses will be supported in schematic, through the netlister, and into T-Spice.

1.6.1.1. Viewlogic Integration Implementation

Viewlogic Integration with Tanner Tools has been developed using two new features, the Tanner Toolbar and the Tanner Project Manager. The Tanner Project Manager assists the user in managing the files associated with a particular project, and allows the user to set the Tanner Working Directory (TWD). The aim of the project manager is to organize all the files associated with a particular design, and should evolve to provide indications to the user on where they are in the design cycle, and what operations need to be done. For example, it might indicate that layout have changed and verification needs to be redone. It also serves to synchronize the files associated with a design. The Tanner Toolbar allows a user to launch tools in the Tanner Tools Pro suite such that file open and

file save operations originate in the TWD. The Tanner Toolbar is intended to be a convenient location where users can access tools in the Tanner Tools Pro suite. Together, these tools will provide a user with a cleaner way to organize several projects. The Tanner Toolbar will be customizable. ViewDraw integration can be accomplished by putting Viewlogic items in the toolbar and replacing the Tanner Project Manager with the Workview Office Project manager.

User Interfaces

The Tanner Toolbar looks and acts much like toolbars in Windows applications. Each button has an icon associated with it. Clicking on a buttons launches the associated application. The first button brings up a popup menu instead of launching an application.

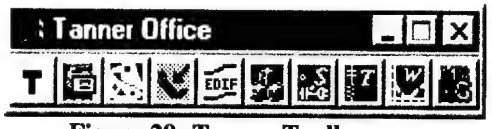


Figure 29: Tanner Toolbar

1.6.2. Mixed Language Models

The goal of this task is to develop a simulation management interface for manipulation of VHDL, VHDL-AMS, SPICE, data-table, "C", and MATLAB oriented models. This interface provides the capability to combine simulations at different levels of abstraction required for final simulation of the entire design. Output of low level simulations in one type may be input into higher level simulations of a different type. The microsystem designer is able to perform top-down design optimization and bottom-up design verification using a variety of integrated simulation techniques and a mix of models described at various levels of abstraction. Tanner has achieved the ability to perform simulations of designs involving models in SPICE, data table, "C", and MATLAB formats and expect to develop additional capability to interface to VHDL and VHDL-AMS simulators in the future.

1.6.3. Matlab Modeling and Mechanical and Thermal Behavioral Simulation

The goal of the Mechanical and Thermal Modeling task is to develop simulation technology capable of processing component models described in terms of the generalized multi-dimensional spring-mass-damper equilibrium equation, or for thermal analysis, in terms of Green's function impulse responses. Tanner has determined that this simulation capability will be most appropriately implemented using the MATLAB interface. The goal of the MATLAB integration task is to develop a general purpose support for evaluating MathWork's *MATLAB™* format models from within the proposed CAD environment.

The foundational component of MATLAB, the high level programming language, is especially well suited for prototyping and testing complex mathematical equations for scientific and engineering applications. Additionally, MATLAB provides a wide variety of specialized algorithms and functions for specific application areas such as signal processing, image processing, wavelets, control systems, etc. The combination of the

MATLAB programming language, mathematical functions, and the application toolboxes provides a wealth of possibilities for developing mechanical, fluidic, thermal, optic, and other types of device models.

Tanner has implemented MATLAB model support, and are planning its release in T-Spice Version 6.5. In this release, the user will create MATLAB models, compile them as DLL's, and then be able to execute them from T-Spice using the T-Spice external model interface. Tanner will create code templates for users, to assist in building the infrastructure code required for communication between the model and T-Spice. These device models may now be incorporated within T-Spice simulations via T-SPICE's MATLAB interface.

1.6.3.1. The Procedure

The procedure for creating a Matlab model of a device, and interfacing the model with T-Spice is described below:

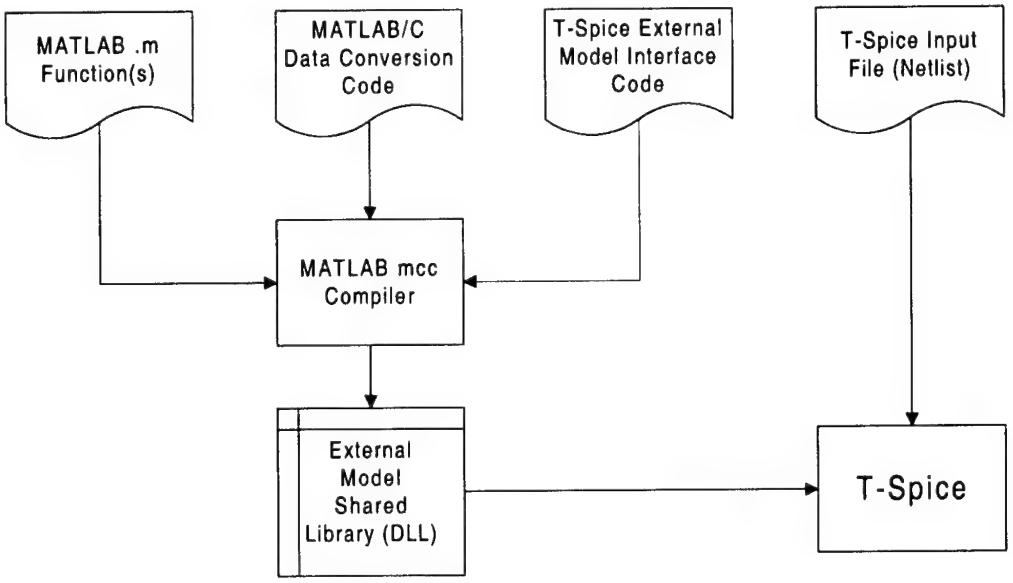


Figure 30: Flow Diagram for T-Spice/MATLAB Interface

i. Develop T-Spice model evaluation and derivative evaluation code using the MATLAB programming language.
ii. Write C code to convert the MATLAB function's input and output data to/from standard C data types.
iii. Modify the T-Spice External Model Interface template (C source code) to call the data conversion and MATLAB model evaluation functions.
iv. Compile the MATLAB functions, Data Conversion code, and T-Spice External Model Interface code using MATLAB's mcc compiler, creating a shared library.
v. Modify the T-Spice input file (netlist) to include a definition of the new external model and one or more instances of the model.
vi. Run T-Spice and view the results with W-Edit.

1.7. Layout Synthesis and Verification

The tasks under this effort will deliver a suite of system level synthesis and verification tools to complement the device level analysis tools. These tools will bring the capability of IC synthesis and verification technology to the domain of the Integrated Microdevice (IM) designer, including Mixed Technology Block Place and Route, Multi-domain Layout Extraction, and Context Sensitive DRC.

1.7.1. Mixed Technology Place and Route

The goal of this task is to develop a Block Place and Route tool capable of placing and interconnecting individual or blocks of integrated microdevices, with a limited set of performance driven constraints. Core code development has also been completed and integrated into the L-Edit layout editor, and the tool has shipped in commercial release.

Highlights:

- Commercial release of Block Place and Route in the MEMS Pro V2 release.
- Demonstration of BPR capabilities using AFRL MEMS-Push design.
- Developed automatic and manual block placement and routing (BPR) capability with the following features:
- Support of industry standard EDIF netlist format.
- Incorporate IP blocks, custom blocks and standard cells into your design, perform basic floorplanning, and evaluate and control interconnect parameters, resulting in a more tightly integrated design.
- Routed blocks may be hierarchically composed into higher levels of design. Blocks may consist of compositions from another BPR cycle, from SPR, from hand layout, or from a library.
- At any time during the design process, easily place and route blocks or perform signal integrity and timing analysis to ensure adherence to performance requirements. Automatic and manual routing may be intermixed with the user-assisted routing feature.
- Persistent connectivity information throughout layout operations.
- Topology-driven automatic routing.
- Built-in fast delay calculator.
- On-line SPICE signal integrity analysis.
- Integrated design environment based on unified physical, electrical, and timing information.
- Block place and route integrated with Standard Cell Place and Route, Subcircuit Extraction, and Layout Editing.
- Prototyped VHDL netlist support.

Development of the Block Place and Route tool came up to speed very quickly, and Tanner has been able to demonstrate manual and automatic placement as well as manual and automatic routing. The culmination of this effort was the shipping of the BPR module

in L-Edit V8 and MEMS-Pro V2. The features of the tool are listed in the above highlights section.

An example of a Q-controlled resonator has been developed and used to test the Block Place and Route tools. Input to the tool consists of a netlist, and a cell library containing the cells referenced in the netlist. Figure 31 below shows the layout of the Q-controlled resonator after the netlist has been read in and the cells have been instantiated in the design. Here the terms cells and blocks are used synonymously. The red lines connecting the cells is the rats nest. The rats nest shows connections using a minimum spanning tree, which connects the terminals of a net such that the length of the net routing is minimized.

Figure 32 shows the layout after the user has performed manual placement, using the rats nest as a guide to minimize congestion. Note that the connection lines stay connected and stretch as block are moved. Figure 33 shows the design after routing has been completed.

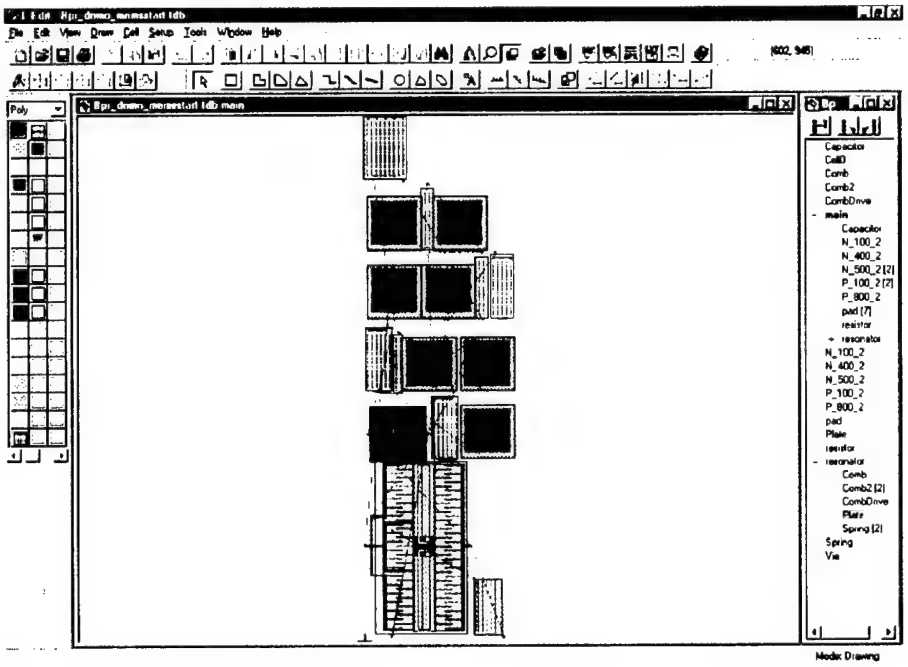


Figure 31: Block Place and Route of Q-Controlled resonator, illustrating initial placement of cells. Red lines connecting cells are the rats nest.

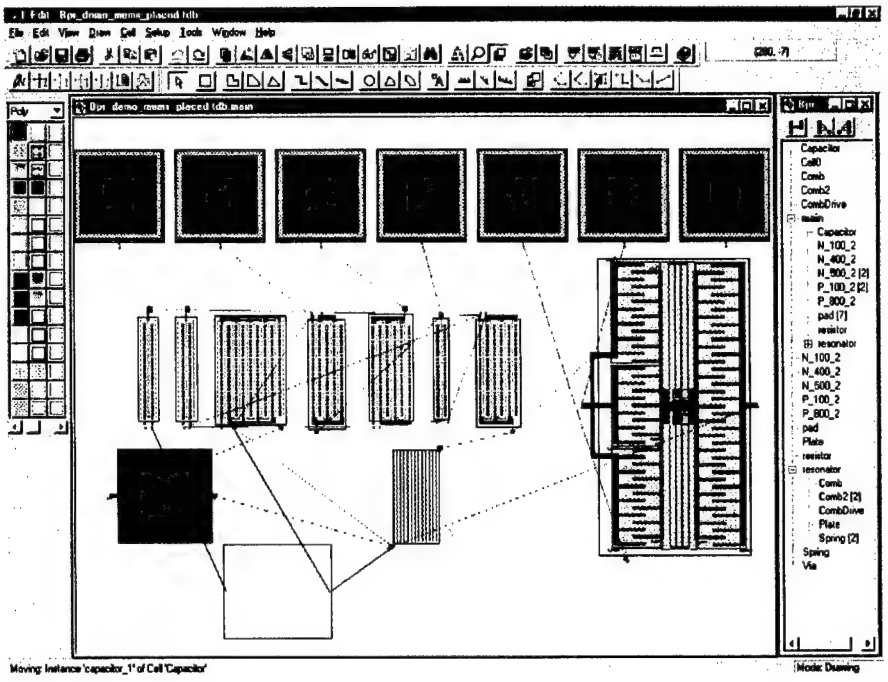


Figure 32: Block Place and Route of Q-Controlled resonator, illustrating manually placed design. Note rubberbanding of connections as block on lower left is moved.

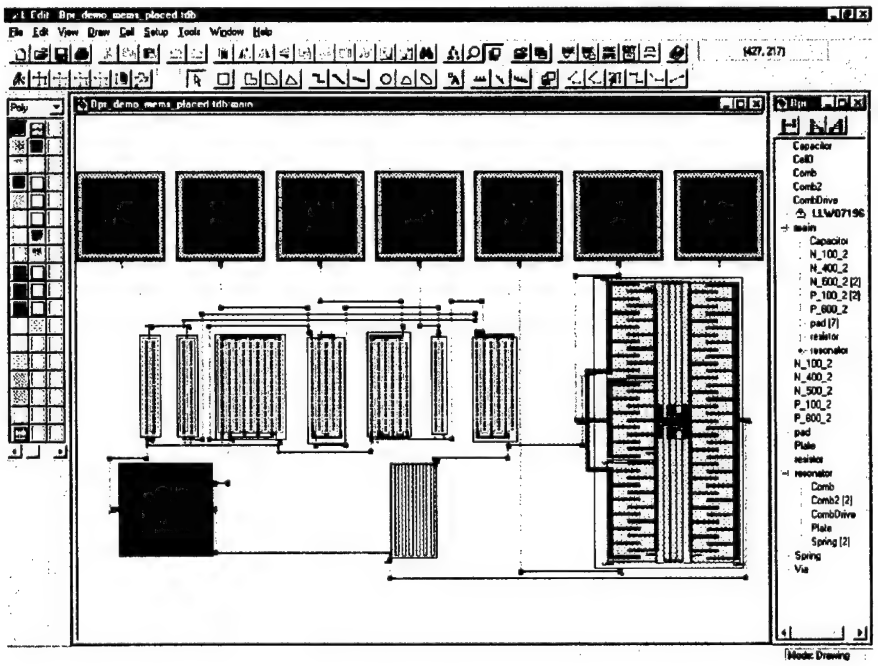


Figure 33: Block Place and Route of Q-Controlled resonator, illustrating routed design.

A delay calculator based on the moment matching method has also been developed and integrated into BPR, for accurate calculation of RLC delay. Nets violating a delay criteria are easily identified. Interactive signal integrity analysis is also integrated into BPR via a convenient user interface to the T-Spice circuit simulator. User are be able to click on any net and quickly obtain results of a simulation be able to probe wave forms at specified test points.

In addition to core Place and Route development, Tanner has investigated support of VHDL netlists in the Block Place and route system. Currently netlist input is through tpr (Tanner Place and Route) and EDIF formats. A partial VHDL parser was developed, and placement and routing of a design was demonstrated, however this feature was not pushed all the way through to commercial release.

The AFRL MEMS-Push chip continues to be Tanner's prime demonstration vehicle for mixed technology BPR applications. Tanner has used SPR to route the CPU, UART, and RTC cores, and used BPR to route the three cores together. Addition of the layout cell and connectivity for the MEMS device will complete the demonstration. Figure 34 shows the placement of the cores in BPR, with the purple lines showing connections between blocks prior to routing. Figure 35 shows the design with routing completed.

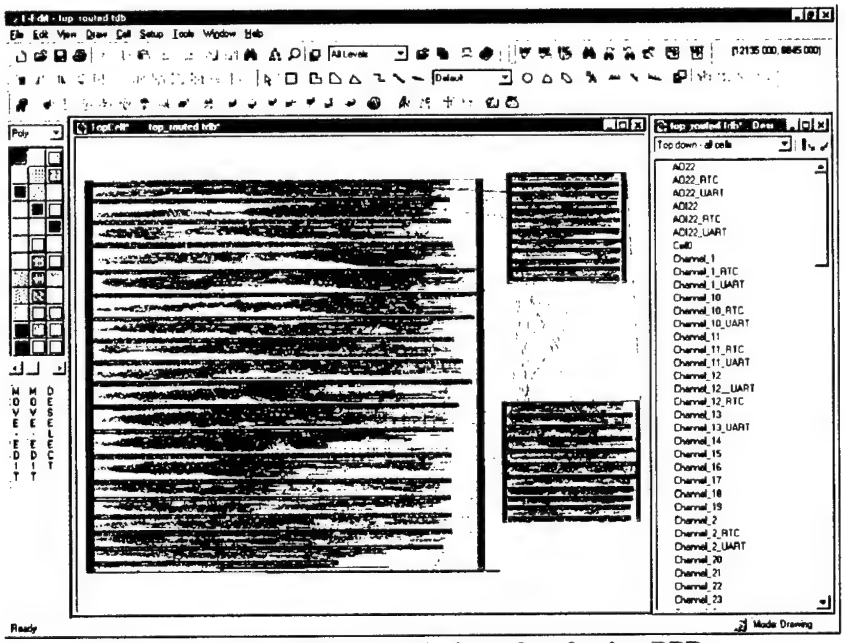


Figure 34: AFRL MES-Push design, placed using BPR.

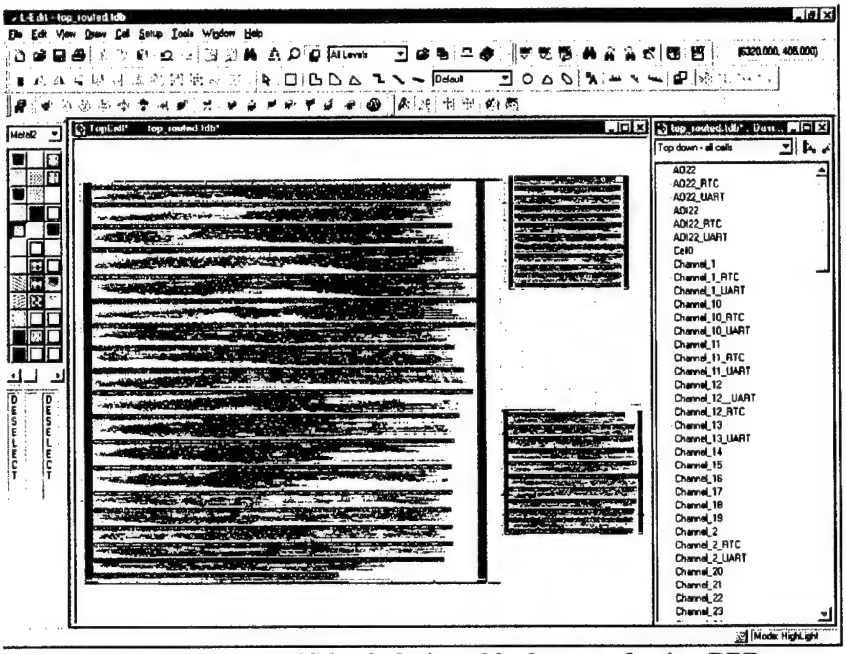


Figure 35: AFRL MEMS Push design, blocks routed using BPR.

1.7.2. Multidomain Layout Extraction

This task involves development and demonstration of a layout extractor capable of extracting mechanical, thermal, and other domain topologies.

Highlights:

- Developed capability to extract MEMS devices from layout. The features builds on a general purpose property tool to attach properties to any object in L-Edit.
- Developed a design of references and equation support for L-Edit properties in support of MEMS device extraction.

Tanner has developed a general purpose property mechanism for objects in L-Edit, and built upon this foundation to support of MEMS device extraction. Properties such as length, width, and number of comb drive fingers may be attached to a cell, as shown in Figure 36. Once properties have been defined on a cell or instance, the extract tool is able to detect these properties and write them into the Spice netlist during extraction. This capability is particularly useful in conjunction with automatic layout macros, as the properties may be coded into the macro, and assigned to the cell during layout generation.

The properties tool has been designed to handle hierarchical assignment of properties. A valuable capability of the inheritance mechanism is that it allows the user to change properties of all instances simply by changing properties of the parent cell, a feature not present in earlier designs of the tool. Cell properties may also be overridden on a particular instance. The Extract tool first inspects the instance for properties, and for any properties that are not found on the instance, Extract will search for on the cell. Instance properties are thus inherited from the cell.

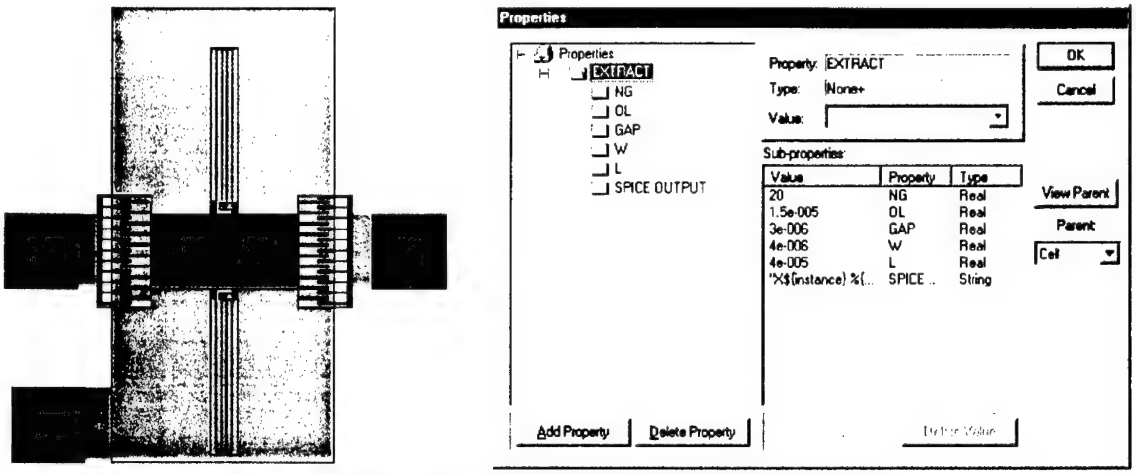


Figure 36: Resonator layout with properties of comb drive.

Tanner has also investigated on how the properties based extract features can be further enhanced. Tanner has worked on designing a mechanism by which properties may reference other properties at any location in the design. Tanner is also designing a mechanism by which properties may contain equations, and these equations may in turn contain references to other properties. In particular, property references will provide the capability for cells to refer to properties on objects within the cell. Since MEMS devices are conveniently extracted as subcircuits, it is important for a cell to be able to communicate with its contents. The ability to reference properties in Extract will provide other unique capabilities. For example, the extractor could reference material properties and process steps in the 3D modeling tool, and calculate the mass of an object. The goal is to design a flexible structure that lets the user customize the configuration to his particular device.

1.7.3. Design Rule Checking

This task involves the development and demonstration of DRC enhancements enabling integrated MEMS/IC design verification, including: i) context sensitive DRC, ii) support of multiple rule sets, and iii) improvements to DRC core engine.

Highlights:

- Developed capability to support multiple rule sets, and apply different rule sets in different areas of the layout.
- Developed capability to automatically partition rules into groups to maximize performance.
- Continued to enhance the fundamental data structures and search engines in support of new curved primitives for MEMS structures.

Context sensitive DRC provides the ability for a set of rules to vary on-the-fly based on the geometry of the layout being verified. Examples of context sensitive DRC are to apply a different width rule to traces exceeding a critical length, or different rules to comb teeth above and below a specified length. Specifically, context sensitive DRC involves

the implementation of a selection feature defined as follows: Select polygons *x* from layer A based on relation with polygons in layer B: *(1) inside, outside, hole; (2) cut, touch, enclose* (with ranges); also selection based on labels, vertex count, number of distinct polygons to which *x* is related. This operation enables context sensitive DRC operations by filtering layout by a specified criteria before continuing processing, either rule checking or additional Boolean operations. Selection operations may be concatenated together. Development of the SELECT feature has been demonstrated, and is ongoing.

Tanner has developed and demonstrated selection operations that permit the construction of logical relations on layers, such as touching, cutting, containment, and edge adjacency. Based upon such relations, one can then select desired sets of objects; for example, one can isolate all objects on layer A that are contained within an object of layer B, or all objects on layer A that cut between two and four objects on layer B. The underlying architecture is sufficiently general to permit the addition of other criteria for the construction of such relations, such as ones dependent upon properties.

The DRC engine permits a second sort of selection - selection of edges based upon metric criteria such as width, length or spacing. As in the case of the logical selection operations, objects selected through DRC operations can then be the subject of successive DRC operations. For example, one may use this form of selection to segregate long and short wires, to which different sets of DRC rules should apply.

An example of All Angle DRC with Selection capability is illustrated in the figure below. On the left is shown a close up of the layout of a motor, with layer Poly 0 in green and layer Poly 1 in red. A ground plate is drawn on Poly 0, and the rotors and stators of the motor are drawn on Poly 1. On the right shown in green is the result of the boolean operation (Poly 0) AND (Poly 1). Also on the right shown outlined in white is the result of the selection operation (SELECT all polygons on Poly 1 that TOUCH TWO polygons on Poly 0). The two polygons on Layer Poly 0 are the ground plate and an additional pad that is out of the figure. This TOUCH rule is able to identify a short stator that has not extended long enough to touch the ground plate, and will thus not function.

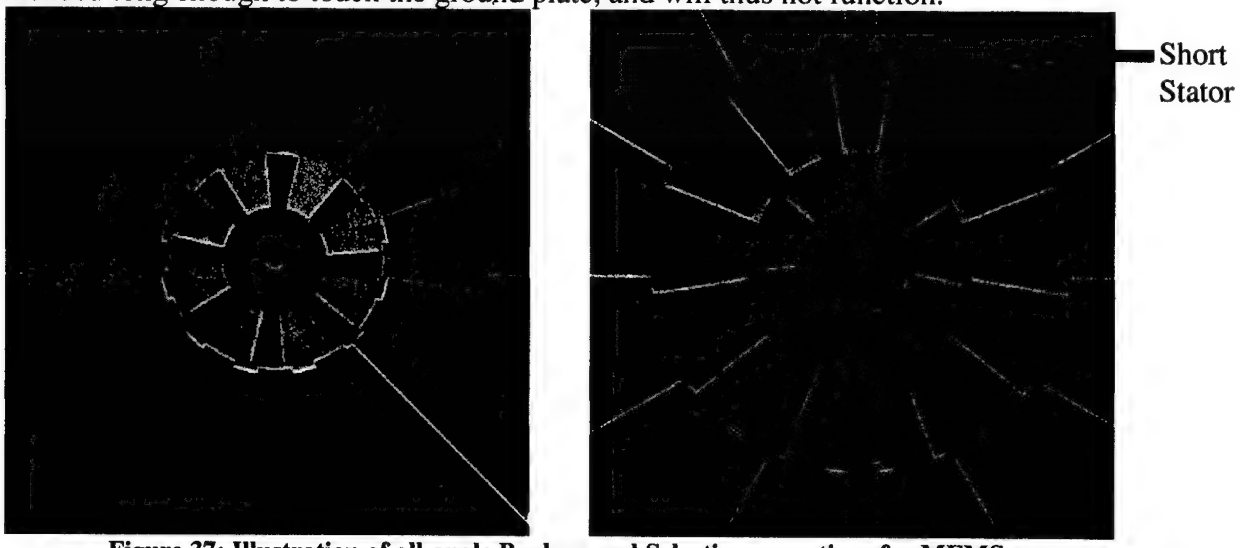


Figure 37: Illustration of all-angle Boolean and Selection operations for MEMS.

Tanner has also developed the feature that enables the user to group rules into different rule sets, and apply those rule sets to different regions of the layout. The rule groups may represent MEMS and IC rules, to be applied separately in the MEMS and IC regions. Rule groups may also be used to apply rules selectively to a region of the MEMS device, as in the case of applying etch hole rules only a plate but not on beams and springs, as shown in Figure 38 to Figure 41 below.

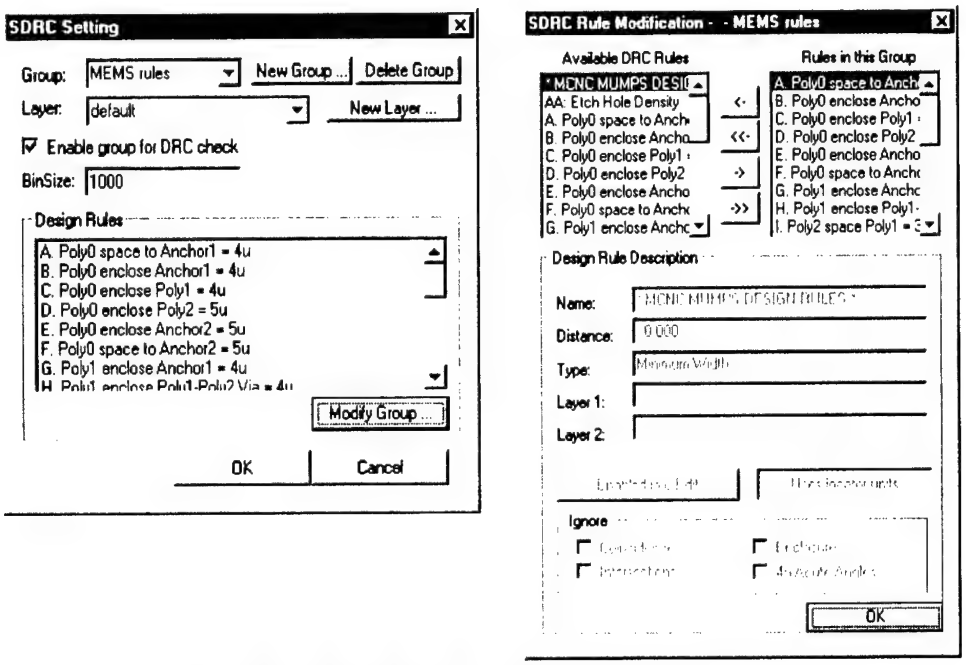


Figure 38: Dialog boxes for multiple rule setup.

```
#
# COPYRIGHT (c) 1999 Tanner Research
#

<Group>
        <Name>MEMS rules</Name>
        <Enable>TRUE</Enable>
        <Layer>default</Layer>
        <BinSize>1000</BinSize>
        <Rules>
                <RuleName>A. Poly0 space to Anchor1 = 4u</RuleName> # grow = 4
                <RuleName>B. Poly0 enclose Anchor1 = 4u</RuleName> # grow = 4
                <RuleName>C. Poly0 enclose Poly1 = 4u</RuleName> # grow = 4
                <RuleName>D. Poly0 enclose Poly2 = 5u</RuleName> # grow = 5
                <RuleName>E. Poly0 enclose Anchor2 = 5u</RuleName> # grow = 5
                <RuleName>F. Poly0 space to Anchor2 = 5u</RuleName> # grow = 5
                ...
        </Rules>
</Group>


<Group>
```

```
        <Name>Etch hole rule</Name>
        <Enable>TRUE</Enable>
        <Layer>EtchHoleCheckArea</Layer>
        <BinSize>1000</BinSize>
        <Rules>
                <RuleName>AA: Etch Hole Density</RuleName> # grow = 18
        </Rules>
</Group>

# END OF FILE
# initial default file
```

Table 7: Sample of multiple rule groups configuration file. First group is applied everywhere, second group for etch holes is applied only on plate.

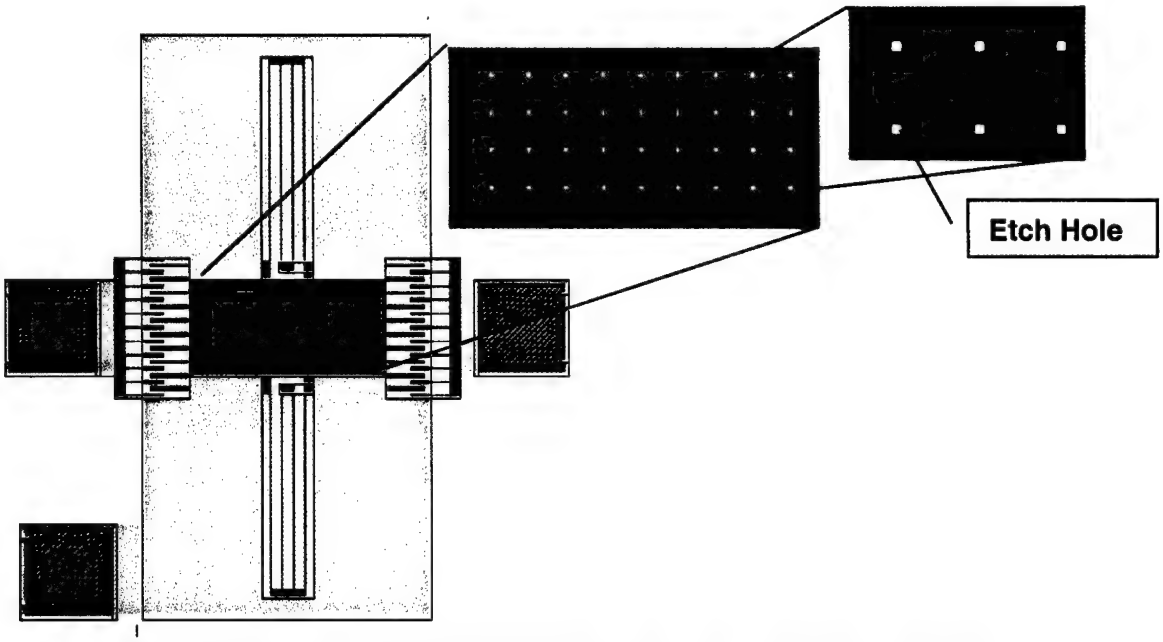


Figure 39: Linear resonator with etch holes in the plate mass. Selectively check for density of etch holes only in plate area of poly1 layer, but not in comb or spring areas.

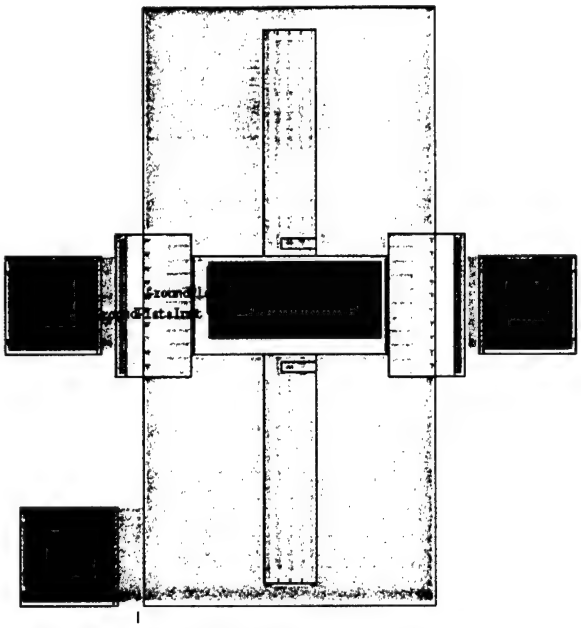


Figure 40: Results of DRC check for etch hole density with rule applied to entire layout. Green color shows violation of etch hole density rule.

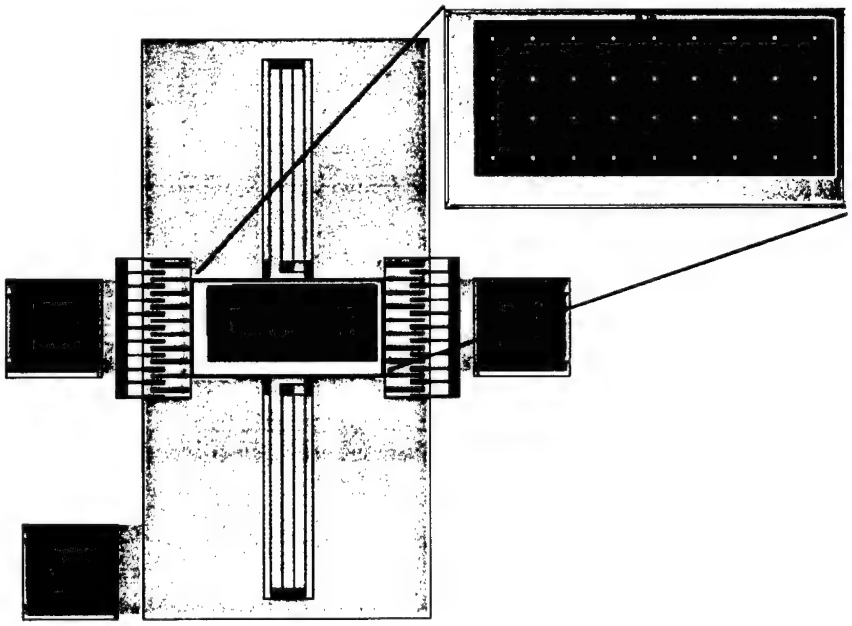


Figure 41: Results of DRC check for etch hole density, with rule applied selectively to plate area. Green color shows violation of etch hole density rule.

Lastly, Tanner has done additional work this quarter to bring curved drawing primitives back into the sweep algorithm for boolean operations and DRC.

1.8. Program Management

This task comprises the following activities: (1) technical and administrative management of the technical data and computer software produced as a result of all work

on the contract, and (2) administration of the contract deliverables, subcontracts and billings.

During this quarter Program Management duties have involved technical direction of the project, attendance and presentation at the Composite CAD PI Meeting in Agoura Hills, and writing of this Final Report.

1.9. Applications Engineering and Productization

This task has involved the development and collection of sample designs, and demonstration of the design process through the system, as well as the management of alpha-site and beta-site releases of the tools. Sample designs developed have been incorporated into tutorials and examples shipped with MEMS-Pro V2. An important aspect of this task is Tanner Research attendance at conferences, symposia, user-group meetings and training classes to identify potential customers, foundry partners and other collaborators. Tanner has actively attended and presented at MEMS industry conferences, and also taught MEMS design as part of the MCNC MUMPS training class.

2. Program Milestones

Tanner has shipped the commercial release of MEMS-Pro V2, which contains the new 3D Modeling module, MEMS libraries and technology setups, the new Block Place and Route module, MEMS Extraction capabilities as well as T-Spice features including the C programmable functional model interface. Also in this release is the ability to easily export 3D models from MEMS-Pro, and import them into ANSYS for FEA analysis.

Many of the capabilities prototyped and demonstrated in this program continue to be developed, and are scheduled to be released in the year 2000. This includes the DRC rule grouping and context sensitive checking features, ViewDraw integration and Schematic features, as well as MATLAB integration. The angled sidewall modeling and analysis features have been prototyped and demonstrated as described in this report. Also, Model Builder was demonstrated at the PI meeting in Agoura Hills, on an accelerometer example. This example, which was created in collaboration with Dr. Selden Crary from the University of Michigan, is also discussed in this report.

Tanner plans on delivering a commercial release of T-Spice Pro V6.5 by the end of Q2 2000, which is scheduled to contain the MATLAB interface, a Viewdraw front end into T-Spice simulation, implementation of the new e and g elements in T-Spice, improved waveform viewing for non-electrical signals, and optimization enhancements. The MATLAB modeling capability and Viewlogic integration have been prototyped, and are now in the commercial development phase.

Physical Design	Date	Accomplished
Demo of Right Angle Solid Model & Visualization	01-Oct-97	Yes
Demo of Layout and Simulation Libraries	01-Dec-97	Yes
Final Specification of Process Definition	01-Apr-98	Yes
Beta of Right Angle Solid Model & Visualization w/GI	01-Jul-98	Yes
Final spec of Library Structure	01-Jul-98	Yes
Demo of Libraries	01-Oct-98	Yes
Alpha of Angled Sidewall Solid Model	01-Jan-99	Yes
Beta of Angled Sidewall Solid Model	01-Jul-99	Yes
Finite/Boundary Element Analysis		
Demo of Automatic 3D FE & BE Meshing	01-Jan-98	Yes
"Thread" through Layout-Solid Model-Mechanical/Ele	01-Jan-98	Yes
Alpha of Integrated (Mech, Elect) Solver w/GUI	01-Jul-98	Yes
Beta of Inegrated Solver (Mech,Elect)	01-Jan-99	Yes
Alpha of Integrated Thermal Solver	01-Jan-99	Yes
Alpha of Harmonic Analysis	01-Apr-99	Yes
Alpha of Angled Sidewall analysis	01-Jul-99	Yes
Beta of Angled Sidewall analysis	01-Oct-99	Yes
Model Builder		
Alpha of Model Builder	01-Apr-99	Yes
Beta of Model Builder	01-Oct-99	Yes
Circuit Refinement Suite		
Schematic Enhancements, MATLAB	01-Oct-98	Yes
Validated multidimensional MEMS Model, VHDL Inte	01-Apr-99	Yes
Full mixed domain model suport	01-Oct-99	Yes
Synthesis and Verification Suite		
Demo of Mixed Tech Block Place and Route	01-Oct-98	Yes
Beta of Multi-domain Extraction	01-Jan-99	Yes
Beta of Context Sensitive DRC	01-Jul-99	Yes

Table 8: Schedule of deliverables.

3. Schedule

The program schedule is shown in Figure 42 below.

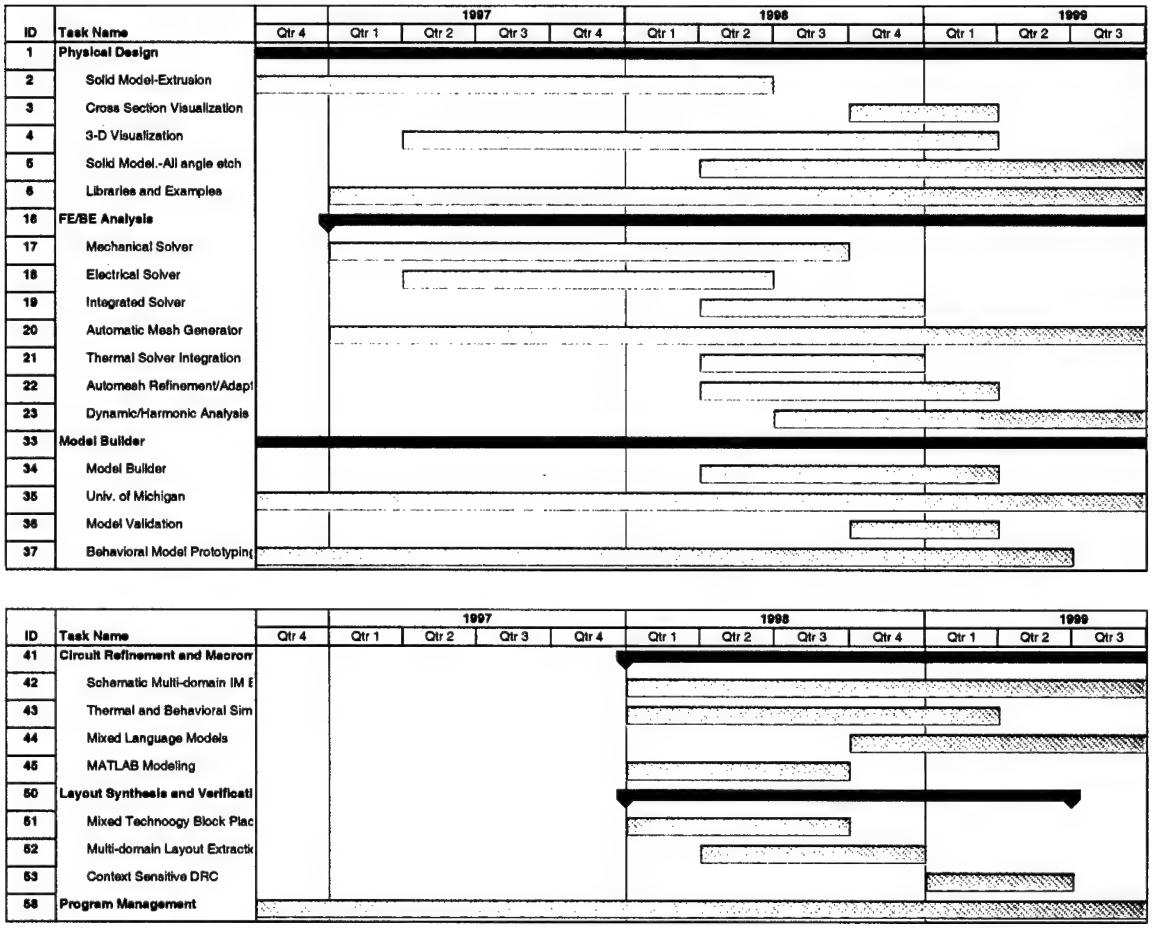


Figure 42: Program Schedule

4. Personnel Status

Table 9 depicts the Team members, email addresses, and their current responsibilities.

Dr. Barry Dyne	barry.dyne @tanner.com	**Program Manager.** Leader for Solid Modeling, FE/BE Analysis
Eugene Chen	eugene.chen @tanner.com	**Contributor.** Schematic.
Dr. Peisheng Gao	peisheng.gao @tanner.com	**Contributor.** 3D Solid Modeling, Visualization.
George Kardaras	Kardaras @tanner.com	**Contributor.** 3D rendering and results postprocessing.

Hee Jung Lee	Heejung.lee @tanner.com	**Contributor.** Process Definition, Behavioral Model Development
Dr. Jian Gong	Jian.Gong@tanner.com	**Contributor:** FEA Mechanical and Thermal Solvers
Dr. David Bernstein	David.bernstein @tanner.com	**Contributor:** Meshing, Self-consistent co-solvers
Dr. Ananth Sethuraman	ananth.sethuraman @tanner.com	**Contributor:** BE EM Solvers
Dr. Mary Ann Maher	maryann.maher @tanner.com	**Task Leader.** Leader for Behavioral Model Builder and Library tasks.
Dr. Scott Wedge	scott.wedge @tanner.com	**Task Leader.** Leader for Schematic and Simulation.
Dr. Harald von Sosen	harald@tanner.com	**Contributor.** Modeling and Simulation
Dr. Alex Khainson	alex@tanner.com	**Contributor.** Mixed technology Place and Route
Dr. Doug Ierardi	ierardi@tanner.com	**Contributor.** Context sensitive DRC
Mr. Nicolas Williams	nicolas.williams @taner.com	**Contributor.** Extract.
Mr. Jin Luo	jin.luo@tanner.com	**Manager:** Application Engineering and Training
Aki Saggio	aki.saggio @tanner.com	**Contract Administration**
Prof. Stephen Senturia	sds@mtl.mit.edu	**MIT Consultant:** Behavioral Modeling
Dr. Selden Crary	crary@umich.edu	**Univ. of Mich. PI:** Model Builder
Rod Morison	rod.morison@tanner.com	**Vice President, Tanner EDA.** Fiscal Review

Table 9: Individuals contributing to the MEMS program.

5. Talks/Presentations

Presented/Published:

- Hee Jung Lee, "Lateral comb drive resonator design uses system-level CAD tools", Micromachine Devices, March 1997, Vol.2, No. 3.
- S.B. Crary, Y.B. Gianchandani, "Parametric Modeling of a Microaccelerometer Using I-Optimal Design of Experiments for Finite Element Analysis", 1996 International Mechanical Engineering Congress and Exposition, Atlanta, GA, Nov. 17-22, 1996.

- Microelectromechanical Systems and the Multiuser MEMS Process Short Course, December 2-4, 1997, MCNC Research Triangle Park, NC. Mary Ann Maher provided instruction on the use of L-Edit for MEMS design to attendees of this three day course on the MUMPS process. Keith Schaefer attended the course to learn more about the MUMPS process, in relation to solid modeling requirements.
- Mary Ann Maher and Hee Jung Lee, "MEMS Systems Design and Verification Tools", SPIE's 5th Annual International Symposium on Smart Structures and Materials Symposium, March 1-5, 1998, San Diego, CA.
- Hee Jung Lee, Harald von Sosen, Mary Ann Maher, "Model Building for Microsystems" First International Conference on Modeling and Simulation of Microsystems, Semiconductors, Sensors and Actuators, April 6-8, 1998 Santa Clara, CA.
- Microelectromechanical Systems and the Multiuser MEMS Process Short Course, April 15-17, 1998, MCNC Research Triangle Park, NC. Mary Ann Maher provided instruction on the use of L-Edit for MEMS design to attendees of this three day course on the MUMPS process.
- Commercialization of Microsystems 98, San Diego, CA. Sept 13-17, 1998. Invited talk by Rod Morison.
- Introduction to MEMS and MUMPs (Short Course). MCNC, North Carolina, June 29-July 1, 1998 and "Nov 16-18, 1998. Instruction on Tanner's MEMS-Pro software presented by Mary Ann Maher.
- UCLA short course on "Smart Structures for Active and Sensory Applications", UCLA, Los Angeles, CA, Aug. 31-Sept. 2, 1998. Contributions by Mary Ann Maher and Ying Xu.
- "Integrated MEMS Design and IC Design CAD Tools", by Hee Jung Lee, F&M 106 (1998) 11.
- "Simulation of MEMS Systems", by Mary Ann Maher, Hee Jung Lee, Micromachining Devices, December 1998. Vol. 3, No. 12
- "Accelerometer and Pressure Sensor DesignUsing MEMS-Pro", by Ying Xu, Patrick Chu, Hee Jung Lee, Mary Ann maher, and Linh Nguyen. Poster presented at the Advanced Microsystems for Automotive Applications, 3rd International Conference, Berlin, Germany, March 18-19, 1999.
- "A Correct-By-Construction Approach to MEMS Design and Analysis" by Barry Dyne, David Bernstein, SPIE Symposium on Design, Test, and Microfabrication of MEMS/MOEMs, Paris, France, May 30 – April 1.
- "Structured CAD Methodology for Integrated MEMS and IC Design" by Linh Nguyen, Hee Jung Lee, Mary Ann Maher, Ying Xu, SPIE Symposium on Design, Test, and Microfabrication of MEMS/MOEMs, Paris, France, May 30 – April 1.

- "Methodology for System Level Simulation, Modeling, and Optimization of MEMS Devices", Linh Nguyen, Hee Jung Lee, Mary Ann Maher, Harald von Sosen, Modeling and Simulation of Microsystems, Semiconductors, Sensors and Actuators, San Juan, Puerto Rico April 19-21,1999.
- Energy-Based Characterization of Microelectromechanical Systems (MEMS) and Component Modeling Using VHDL-AMS", A. Dewey, H. Dussault and J. Hanna, E. Christen, G. Fedder, B. Romanowicz, and M. Maher, Modeling and Simulation of Microsystems, Semiconductors, Sensors and Actuators, San Juan, Puerto Rico April 19-21,1999.

*U.S. GOVERNMENT PRINTING OFFICE: 2002-710-038-10213

MISSION
OF
AFRL/INFORMATION DIRECTORATE (IF)

The advancement and application of Information Systems Science and Technology to meet Air Force unique requirements for Information Dominance and its transition to aerospace systems to meet Air Force needs.